University of Sheffield

Doctoral Thesis

Blind Image Deconvolution using Approximate Greatest Common Divisor and Approximate Polynomial Factorisation

Author:

Hanan Halawani

Supervisor:

Dr. Joab Winkler

A thesis submitted in fulfilment of the requirements

for the degree of Doctor of Philosophy

in the

Machine Learning group

Computer Science Department

March 2018

.

UNIVERSITY Of SHEFFIELD

Abstract

Faculty of Engineering
Computer Science Department

Doctor of Philosophy

Blind Image Deconvolution using Approximate Greatest Common Divisor and Approximate Polynomial Factorisation

by Hanan HALAWANI

Images play a significant and important role in diverse areas of everyday modern life. Examples of the areas where the use of images is routine include medicine, forensic investigations, engineering applications and astronomical science. The procedures and methods that depend on image processing would benefit considerably from images that are free of blur. Most images are unfortunately affected by noise and blur that result from the practical limitations of image sourcing systems. The blurring and noise effects render the image less useful. An efficient method for image restoration is hence important for many applications.

Restoration of true images from blurred images is the inverse of the naturally occurring problem of true image convolution through a blurring function. The deconvolution of images from blurred images is a non-trivial task. One challenge is that the computation of the mathematical function that represents the blurring process, which is known as the point spread function (PSF), is an ill-posed problem, i.e. an infinite number of solutions are possible for given inexact data. The blind image deconvolution (BID) problem is the central subject of this thesis. There are a number of approaches for solving the BID problem, including statistical methods and linear algebraic methods. The approach adopted in this research study for solving this problem falls within the class of linear algebraic methods. Polynomial linear algebra offers a way of computing the PSF size and its components without requiring any prior knowledge about the true image and the blurring PSF.

This research study has developed a BID method for image restoration based on the approximate greatest common divisor (AGCD) algorithms, specifically, the approximate polynomial factorization (APF) algorithm of two polynomials. The developed method uses the Sylvester resultant matrix algorithm in the computation of the AGCD and the QR decomposition for computing the degree of the AGCD. It is shown that the AGCD is equal to the PSF and the deblurred image can be computed from the coprime polynomials.

In practice, the PSF can be spatially variant or invariant. PSF spatial invariance means that the blurred image pixels are the convolution of the true image pixels and the same PSF. Some of the PSF bivariate functions, in particular, separable functions, can be further simplified as the multiplication of two univariate polynomials. This research study is focused on the invariant separable and non-separable PSF cases.

The performance of state-of-the-art image restoration methods varies in terms of computational speed and accuracy. In addition, most of these methods require prior knowledge about the true image and the blurring function, which in a significant number of applications is an impractical requirement. The development of image restoration methods that require no prior knowledge about the true image and the blurring functions is hence desirable. Previous attempts at developing BID methods resulted in methods that have a robust performance against noise perturbations; however, their good performance is limited to blurring functions of small size. In addition, even for blurring functions of small size, these methods require the size of the blurring functions to be known and an estimate of the noise level to be present in the blurred image.

The developed method has better performance than all the other state-of-the-art methods, in particular, it determines the correct size and coefficients of the PSF and then uses it to recover the original image. It does not require any prior knowledge about the PSF, which is a prerequisite for all the other methods.

Acknowledgements

Firstly, all praise and thanks go to Allah (God) for helping me in my PhD journey. I want to deeply thank all those who stood by me and who offered endless help and support. This work would not have been achieved without the great love, help, support, and prayers of my two adorable parents, my father (Talal) and the greatest love in my world, my mum (Enam). Thanks to both of them. This piece of work was my father's dream for a long time as he had been offered a scholarship to finish his studies, but for our sake and because myself and my brothers were too young, he did not go and instead sacrificed his whole life to teach us. Here, I am making his dream come true. Mum and Dad, all my love and thanks.

I am also very grateful to my supervisor (Dr. Joab Winkler) for all his continuous support, encouragement, and guidance. Lots of thanks go to my university (Najran University) for offering me this scholarship and funding to get my PhD.

I want to deeply thank many people who have provided me with a lot of help, starting with my beloved husband (Hashiam) who has encouraged me to finish my PhD and who thankfully agreed to stay by himself at home. My utmost gratitude goes to my two stunning kids Ahmed and Ibrahim (both were born during my study period). They tolerated their mum's involvement with her studies.

I am very grateful to my brothers and all my family members whose encouragement was endless.

Last but not least, my thanks and love go to my friends who were just like my sisters, and who made me feel like I was at home.

Finally, I am grateful and so thankful to my Saudi Government for giving us this chance to continue our studies.

Contents

List of Figures

List of Tables

List of Abbreviations

AM	...	**A**lternating **M**inimization algorithm
AGCD	...	**A**pproximate **G**reatest **C**ommon **D**ivisor
APF	...	**A**pproximate **P**olynomial **F**actorisations
BID	...	**B**lind **I**mage **D**econvolution
FFT	...	**F**ast **F**ourier **T**ransform
DFT	...	**D**iscrete **F**ourier **T**ransform
GCD	...	**G**reatest **C**ommon **D**ivisor
IBD	...	**I**terative **B**lind **D**econvolution
IFFT	...	**I**nverse **F**ast **F**ourier **T**ransform
IDFT	...	**I**nverse **D**iscrete **F**ourier **T**ransform
LP	...	**L**inear **P**rogramming
LU	...	**LU** decompositions
LSE	...	**L**east **S**quares **E**rror
LS	...	**L**east **S**quares
MAP	...	**M**aximum **A** **P**osteriori
MSE	...	**M**ean **S**quared **E**rror
MAE	...	**M**ean **A**bsolute **E**rror
ML	...	**M**aximum **L**ikelihood Restoration Method
NP	...	**N**oise **P**ower
PNSR	...	**P**ower of **N**oise-to-**S**ignal **R**atio
PSNR	...	**P**eak **S**ignal-to-signal **N**oise **R**atio

PSF	...	**P**oint **S**pread **F**unction
PDF	...	**P**robability **D**ensity **F**unction
PAM	...	**P**rojected **A**lternating **M**inimization
QR	...	**QR** decomposition
RMSE	...	**R**oot **M**ean **S**quared **E**rror
SNTLN	...	**S**tructured **N**on-linear **T**otal **L**east **N**orm
SVD	...	**S**ingular **V**alue **D**ecomposition
SNR	...	**S**ignal to **N**oise **R**atio
TVBID	...	**T**otal **V**ariation **B**lind **I**mage **D**econvolution
TSVD	...	**T**runcated **S**ingular **V**alue Decomposition
VD	...	**V**ariational **D**irichlet Approximation
ICA	...	**I**ndependent **C**omponent **A**nalysis
BCC	...	**B**lind **S**ource **S**eparation

List of Symbols

$\mathcal{F}$	...	Exact image
$\mathcal{G}$	...	Blurred image
$\mathcal{P}$	...	Point spread function array
$\mathcal{N}$	...	Additive noise
$\mathcal{E}$	...	Measurement error
$F, G, P, \ldots$ elc	...	Matrix form of $\mathcal{F}$,$\mathcal{G}$,$\mathcal{P}$,$\ldots$ elc
$\mathbb{F}, \mathbb{G}, \mathbb{P}, \ldots$ elc	...	the $\mathcal{F}$,$\mathcal{G}$,$\mathcal{P}$,$\ldots$ elc in the frequency domain
frequency domain $S(f,g)$	...	Sylvester resultant matrix for the polynomials $f(x)$ and $g(x)$
$S_k(f,g)$	...	Sylvester subresultant matrix of order k for the polynomials $f(x)$ and $g(x)$
$f(x)$	...	exact form of the polynomial
$\hat{f}(x)$	...	inexact form of the exact polynomial $f(x)$
$\tilde{f}(y)$	...	preprocessed form of the polynomial $f(x)$
θ	...	a scale factor
θ_{o}	...	the optimal value of θ
α	...	a scale factor
α_{o}	...	the optimal value of α
$\lVert\cdot\rVert$	...	$\lVert\cdot\rVert_2$

The thesis is dedicated to my parents Talal and Enam, my aunt Badia, my husband Hashiam, my son Ahmed and my son Ibrahiam, without whom none of my success would be possible.

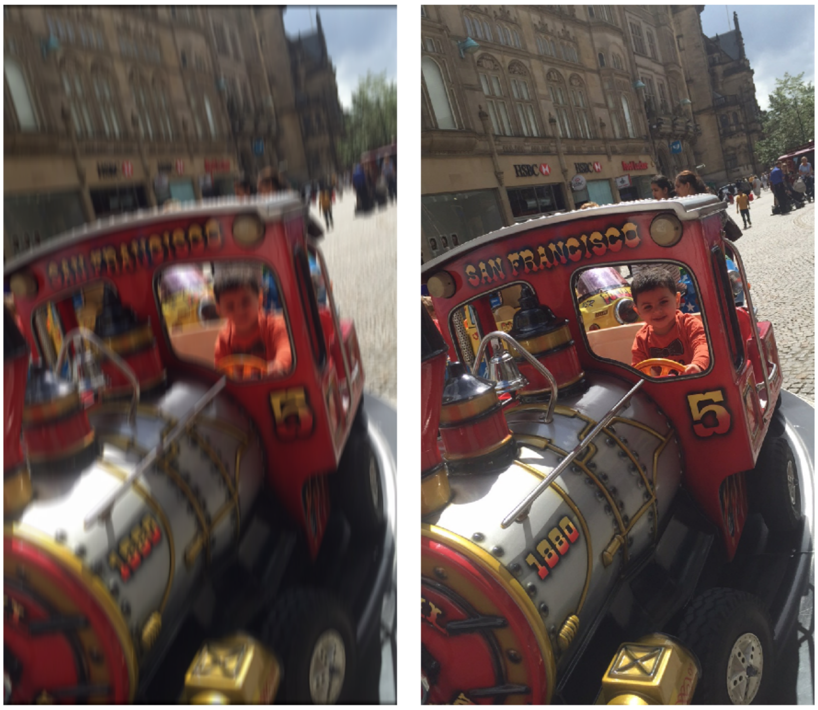
Chapter 1

Introduction

1.1 Background

Images, in modern science and in everyday life, are ubiquitous and indispensable. It is natural to utilize the abilities of our visual system to display our knowledge of the world in graphical form. Images occur in photography, medical imaging, remote sensing, astronomy and microscopy. In each case, there is an essential scene or object that we wish to portray; the image is a visual representation of this knowledge.

Unfortunately, imaging processing, as with any other information extraction processing, is never perfect. Image degeneration occurs in the form of blur, noise, and other degradations in the recorded image due to imperfections in the capturing and imaging processes. Thus, the image emerges as a degraded version of the original image. This degeneration results from various sources such as optical imperfections in the case of a digital camera (e.g. lens defocus) or atmospheric blurring in the case of aerial/satellite photography, etc. [10, 11]. Failure to focus the camera properly or camera shake can both lead to blurred images being generated. Sometimes the lost data may contain information we are interested in. It may be valuable to try to restore these hidden details in order to interpret the underlying scene that generated this information. Figure 1.1 shows an example of a blurred image. Blurred images are a frustrating sight to anyone



(a) A blurred Photographic image

(b) What we really want to see

Figure 1.1: Blurred and original images: example demonstrating motion blur due to camera shake.

who has tried taking photographs for a scientific purpose, for example, image interrogation for feature extraction. Users usually are frustrated that an important proportion of these photographs may end up being unclear, although there have been recent advances in technology. Of course, sometimes these photos may be retaken in the hope that the next photo will be clear, but frequently they are of a unique scene that can only be taken once. It would be beneficial if these scenes could be recovered and their details restored.

Nowadays, cameras have been developed that can handle the imperfections of improper focusing and camera shake by fitting them with image stabilization, auto-focusing, sensitive image sensors and anti-camera shake mechanisms. The goal behind these technologies is to remove blur [11]. However, such cameras are heavier, have high energy consumption and cost more. Despite the camera industry providing these solutions, camera shake blur has only been addressed in a limited manner [11].

Recently, some of the problems of image processing have been overcome by signal processing techniques [11, 12], via post-processing these blurred and noisy images. They estimate the blurred images to an acceptable level. By representing the image restoration process mathematically, and applying prior information about the blurred images, restoration methods such as blind image deconvolution (BID) can be performed to recover detail and reduce image noise.

The rest of this chapter will consider the background to the image restoration problem, including how and why it is a significant problem. A more detailed study will be proposed in this chapter, including a presentation of the problem formulation and solution methodologies. The chapter concludes with an outline of the rest of the thesis.

1.2 Image Restoration and Blind Deconvolution

The image restoration problem pertains to the linear filtering of convolved signals and it can be dated back to the early 1960s [13]. Image restoration refers to a process that recovers the accurate and true image from its distorted version using limited or prior information about the degradation phenomenon. It seeks to estimate and reconstruct a sharp image from the blurry image to improve accuracy. Given a blurred image and the blurring function, the original image is detected by solving an inverse problem called *deconvolution*. The image restoration problem is a non-trivial task because it is ill-posed, which means that the number of possible solutions is infinite for given data.

Image restoration may be classified, depending on blur, into two classes [14, 15]: non-blind image deconvolution, in which the blurring function, or as it is called, point spread function (PSF) is known during the extraction of the true image from the degraded one, and blind image deconvolution, in which the blurring function and true image are unknown. In the first situation, there are various well-known deconvolution method techniques, for instance, least-squares (LS) filtering, recursive Kalman filtering, Wiener filtering, inverse filtering, and constrained iterative deconvolution methods [14, 16–19].

Unfortunately, non-blind image deconvolution methods that assume a known blurring function are not suitable for many real imaging application processes [16]. This is because in practical situations little knowledge about the true image is available and information about the blur is also unknown. There has been significant work on the restoration of astronomical images and current discussion by researchers is still ongoing [11]. There have been applications in computer vision [20], remote sensing, medical imaging [21], [22], but the interest of law enforcement agencies in this area has increased [11]. All these applications require accurate images and any small change in the image content can lead to incorrect information during the analysis process.

Consequently, image restoration solutions can be provided by BID, which can be used to recover handheld camera photographs and videos [23, 24], remote sensing data [25], aerial and satellite photography [26], biomedical images [21, 27], industrial tomography images [28], astronomical photos [29], audio and seismic signals [30], and other sources of images and signals as well.

Over the years, many BID methods and restoration filters have been developed to estimate an acceptable image. These methods use the spatial domain or the frequency domain. In addition, they range from separate or simultaneous of blurring function estimation techniques, and parametric to non-parametric techniques. Examples of these methods are the Richardson-Lucy method [6], total variation blind deconvolution [31], the maximum likelihood (ML) method [16, 32], minimum entropy deconvolution (MED) [33], non-negativity and support-constraint recursive inverse filter (NASRIF) [34], simulated annealing [35], and multi-channel blind deconvolution [35].

Although these methods achieved satisfying solutions to some BID problems, they are lacking in terms of restoration quality, computational efficiency and robustness. Actually, when real-life distorted images are to be restored, the robustness of these methods becomes questionable. The reason behind this is that the actual deconvolution process suffers significantly from deblurring noise and ringing effects. The noise and ringing disturbs the restoration process by changing the statistical properties of the image data; consequently, the removal of blur from an image will be inefficient. These effects can be caused by sensitivity of the recording device to different environmental factors resulting in an instant poor lighting condition [11]. Moreover, it has been proved that ringing effects occur due to errors in the estimation of the blurring function [36]. Therefore, the blurred image cannot be efficiently modelled by the usual parametric blur methods [23, 37]. Most of the BID techniques focus on calculating and removing the blur from a distorted image. The rest of this chapter summarizes the computational challenges of BID, clarifying why this computation is a non-trivial task and hence underlies the motivation for the work presented in this thesis.

1.3 Computational Challenges of Blind Image Deconvolution

Computing the true image of a degraded image poses the most difficult problem for image processing owing to the degraded image being expressed as the convolution of two components: the true image with PSF. Thus, the estimation of the original image process must be unique for deconvolution: this means that it can determine which component belongs to the original image and which to the PSF. Moreover, in the BID process there are practical constraints that must be considered when restoring an image:

- Generally, BID is an ill-posed problem. The ill-posed problem means that there may not be a unique solution for image restoration. Regarding the partial information about the image that is used to formulate an optimality criterion, there will be various different estimates of the PSF with the true image that may lead to an optimal solution. The challenge here is to find the appropriate procedure and construct additional assumptions on the imaging system or BID methods that will contribute towards choosing the optimal solution.

- BID is a difficult problem because the blurring function converts the high spatial frequency component of the original image into the lower spatial frequency component of the blurred image, as is clearly shown in Figure 1.2. This means that the PSF is a low-pass filter that must be known in order to restore the original image. The essential problem in BID is the computation of the correct PSF when only the blurred image is provided.

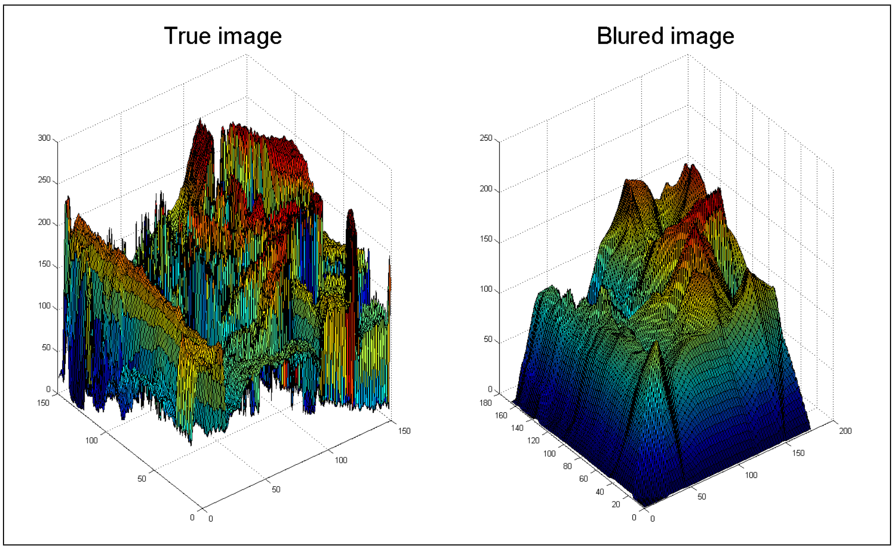


FIGURE 1.2: The component of the original image and blurred version of the image.

- Artificial blurred images are produced by convolving a PSF with a real image see Figure 1.3. This process results in a blurred image that has a bigger size than the real image. The added pixels on all the edges are called the boundary of the blurred image. Normally, natural blurred images are provided without these boundaries. This situation, of missing boundaries, presents a challenge in the deconvolution process of natural images. The size of these boundaries depends on the type of the convolution (linear/circular) and the size of the PSF. In the circular convolution case the resulted blurred images have no boundaries. The linear convolution results in blurred images that have boundaries of size that depends on the size of the PSF, however these boundaries are missing from the naturally blurred images, see Figure 1.4.

It is crucial to consider these constraints when developing the BID algorithm. Although there have been many recent research studies and many existing BID methods, no method in the literature has been suggested that can absolutely claim to be the

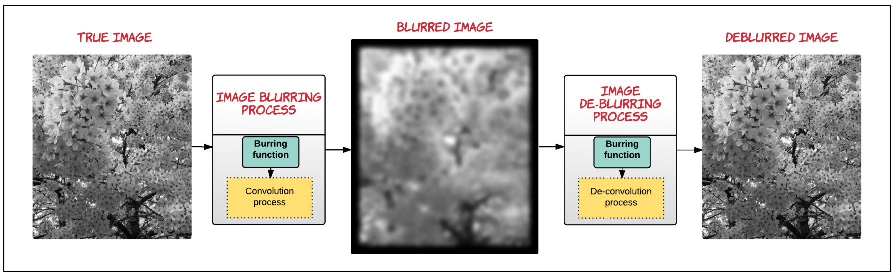


Figure 1.3: Artificial image convolution and deconvolution processes

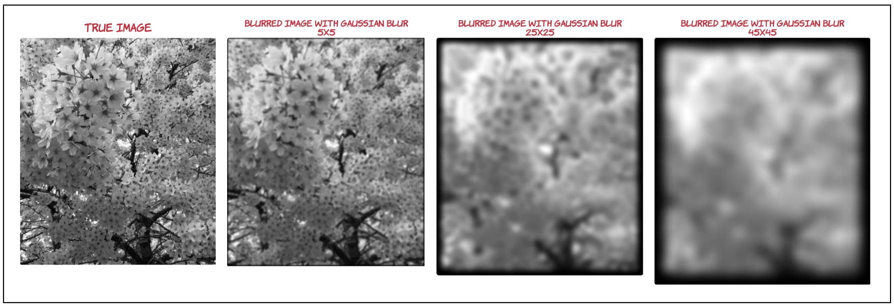

TRUE IMAGE
BLURRED IMAGE WITH GAUSSIAN BLUR 5X5
BLURRED IMAGE WITH GAUSSIAN BLUR 25X25
BLURRED IMAGE WITH GAUSSIAN BLUR 45X45

Figure 1.4: Boundary size of blurred image depends on size of the PSF

solution for the BID problem [12]. So far, the literature has shown that there is no satisfactory method that can suppress the noise significantly while preserving true image details effectively [16]. Furthermore, determining the size of the PSF is the hardest part in PSF estimation and image deblurring. The literature survey shows that most of the deblurring methods assume that the PSF is known and hence the size of the boundaries is also known. The boundary missing information of pixels located outside the degraded image is needed to solve the deconvolution problem. Not all the deblurring frameworks require the missing boundary information. Some of these methods assume known PSFs while others require priors to be set and others put some assumptions on the source images, for more details see Chapter 2. Within the framework of this research a preprocessing stage of naturally blurred images is required. In this stage the blurred image is processed in order to estimate the boundaries of the blurred image before processing with the developed method, see Figure 1.5. To estimate the boundaries of the blurred image, different extrapolation methods could be used. These methods need to be investigated in order to select the one with the best performance. If the extrapolation is good then the algebraic geometry method will work well once the problem of the algebraic method is solved. Hence, the focus of this research is on solving the algebraic deconvolution method. The research scope and objectives of this research study are illustrated clearly in the following sections.

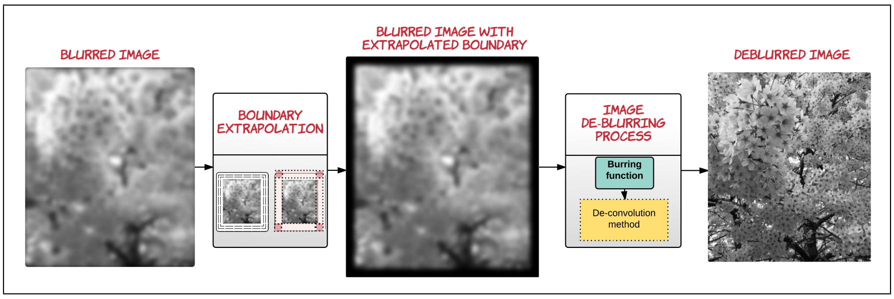


FIGURE 1.5: Proposed natural image deconvolution process

1.4 Research Scope

This research seeks to develop a method for the solution of the problem of BID in which a blurred image is formed by spatially invariant and parametric blur. The spatially invariant blur includes both separable and non-separable PSFs. The original image is processed by a full 2D convolution of spatially invariant blur. The parametric PSFs means that the blurring function can be easily described using a functional or parametric form.

The blurred image is formed by the convolution of the exact image and PSF, therefore, in some cases the blurred image is larger than the exact image. However, in circular convolution the exact image and blurred image are of the same size. In this research study, it is assumed that the blurred image is larger than the exact image. This assumption is made because this study is an investigation of polynomials based method, and in particular, approximate polynomial factorisation, to solve the problem of BID. Any errors in the results are therefore due to the developed method, and not the effects of approximation due to the boundary conditions.

This research will not focus on natural image deconvolution, this problem needs to be addressed in future work. Also, in this research spatially variant blur and non-parametric (arbitrarily shaped) blurs are not considered.

1.5 Research Purpose

This research seeks to remedy the BID problem by developing an efficient algorithm for deblurring degraded images, with focus on the following factors: robustness in the estimation of separable and non-separable PSFs, estimation performance; and computational efficiency.

1.6 Research Objectives

The central objective of this research, under perfect boundary conditions, is to devise an efficient algorithm for BID problem, such that prior knowledge of the blurring function (PSF) and exact image are not required. The proposed algorithm is designed to incorporate the identification of PSF and the restoration of the image in two separate computational algorithms. The first algorithm computes the blur function from a blurred image, and is then used in the second algorithm to remove the blur from the blurred image. This will be done by representing the blurred and deblurred images, and the PSF as bivariate polynomials in which the coefficients are the pixel values. The PSF can be formulated as the approximate greatest common divisor (AGCD) of two degraded images that are represented by two bivariate polynomials. This research is carried out using approximate polynomial factorisation (APF) computational algorithms using a structure-preserving Sylvester resultant matrix method of two polynomials to compute a robust AGCD. The proposed work differs from other work in that it does not require any prior knowledge to recover an image from its blurred version.

1.7 Research Contributions

The aim of this research is to develop a new BID algorithm based on AGCD. Although the AGCD computation has been previously employed for solving the BID problem, the developed method differs in many aspects from the existing AGCD based method (Pillai and Liang [18, 19], Li et al. in 2010 [1, 2] and Danelakis et al. [38]).

Pillai and Liang [18, 19] put forward the initial ideas of using algebraic method for solving BID problem. Their application of algebraic method used Greatest Common Divisor (GCD). They used the stander Sylvester resultant matrix to compute the GCD which does not work in the presence of noise. Li et al. in 2010 [1, 2] developed an Approximate GCD (AGCD) using Bezout matrix and the Fast Fourier Transform (FFT). Although this method performs better than Pillai and Liang [18, 19] GCD method in terms of noise sensitivity, however, it is still sensitive to noise caused by the structure of the Bezout matrix. Danelakis et al. [38] developed an AGCD based method using the updated form of Sylvester resultant matrix for solving BID problem caused by separable three-by-three PSF blurring functions.

This research developed algorithms for BID problem based on AGCD. The developed algorithms used a structure-preserving Sylvester resultant matrix method to compute the AGCD. In particular, QR decomposition is used to compute the degree of the AGCD. This method of the degree computation is more efficient and accurate than the above existing methods. The coefficients of AGCD which represent the PSF pixels values are computed using an Approximate Polynomial Factorisation (APF) method. The use of structure-preserving Sylvester matrix in computation of APF and QR resulted in a more robust and efficient method for computation of the AGCD. The developed method is more robust against noise than the above existing method with noise levels that are relatively high in algebraic methods. However, these noise levels are relatively small in comparison to noise levels that are normally assumed in BID image processing method.

In addition to the above contributions, the following contributions have been made in this research study:

- This research has developed a new BID method using Approximate Polynomial Factorisation (APF) algorithm based on structure-preserving Sylvester matrix.

- The developed method computes the PSFs from blurred images without prior knowledge about the PSF which means the developed method is totally blind. The PSF is computed in two separable stages. In the first stage, the size of the PSF is computed using QR decomposition, and then followed by the second stage in which the PSF component values are calculated. The computed PSFs are then used in a polynomial division to compute the deblurred images.

- The developed method is applicable for computing the PSF of various sizes.

- The developed method is applicable for computing the PSF regardless of whether the PSF is separable or non-separable.

The following Figures 1.6, 1.7 and 1.8 are examples of the results obtained using the developed method. All experiments in this thesis are preformed under perfect boundary conditions. In Figure 1.6, the original image of 180 × 180 pixels has been blurred by a separable PSF with size of 25 × 25 pixels and a small error of 10^{-7} was added to the original PSF. In addition, a relatively small additive noise of 10^{-8} was added to the result of the convolution of the blurred image. The result of the convolution process is a perfect blurred image (perfect boundary). Figure 1.6 shows that the developed method is able to restore the exact image without having any knowledge about the size of the PSF. In Figure 1.7, the original image of 180 × 180 pixels has been blurred by a non-separable PSF with a size of 15 × 15 pixels and an error of 10^{-8} is added to the PSF. A small additive noise of 10^{-6} has then been added to the blurred image. In Figure 1.8 the original image of 180 × 180 pixels is blurred by non-separable Gaussian blurring PSF and non-separable Motion blurring PSF of size 27 × 17 pixels in order to generate the two distorted versions of the images. The figures show the results that are obtained from the developed method. It is clear that the developed method is able to restore the exact image, with large size of PSF, without having any knowledge about the PSF.

More information about the developed method is discussed in Chapters 4, 5 and 6. The claim about additive noise has been exempted in the experiment of Chapters 4 and 5 because the additive noise is relatively small compared with image processing field.

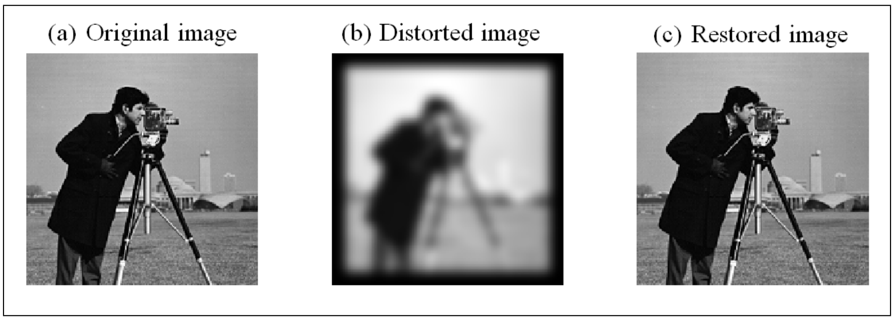


Figure 1.6: Example of BID problem with a separable PSF is solved by developed method.

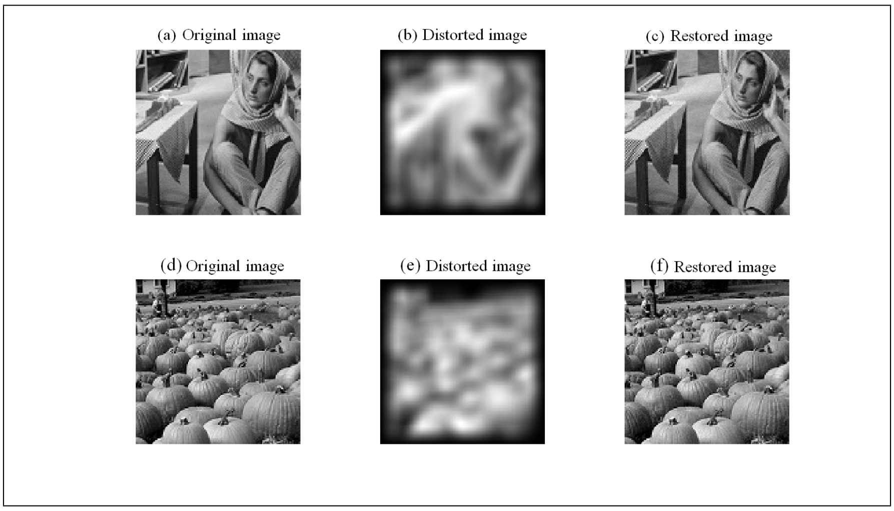


Figure 1.7: Example of BID problem with a non-separable PSF is solved by developed method.

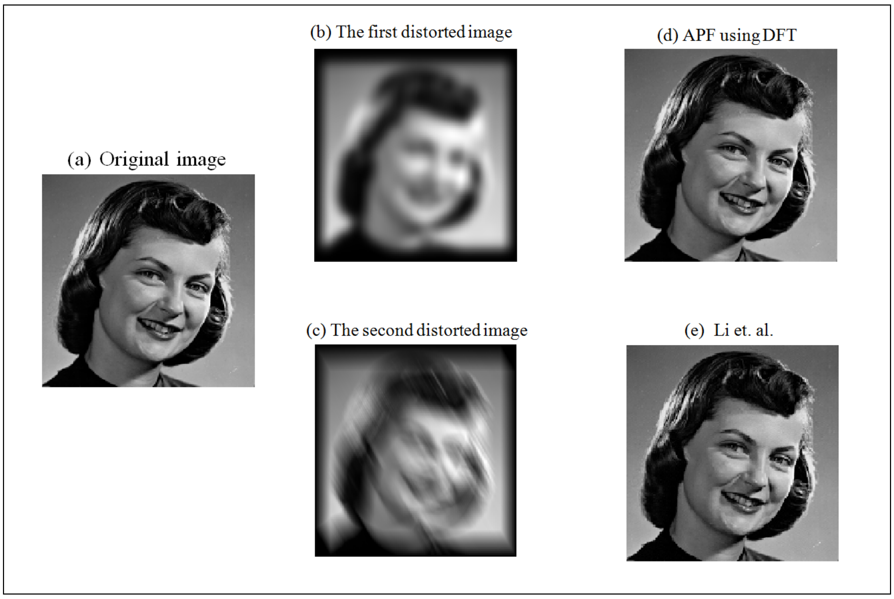


Figure 1.8: Example of BID problem with a non-separable PSF is solved by (d) developed method based on APF and DFT, and (e) the method developed by Li et al. [1, 2].

However, The claim about additive noise is considered in the experiment of Chapters 6 to examine the behavior of two different BID methods based on linear algebra in the presence of noise.

1.8 Research Structure

The rest of this thesis is divided into seven chapters.

Chapter 2 introduces the BID problem formulation and provides fundamental information about the problem. This is followed by a description of the procedure of blurring an image, including a definition and some properties of the blurring function, and the most commonly occurring types of blur. Then, some image restoration filters are detailed, beginning with basic image restoration filters followed by the most recent approaches. Functionalities, advantages, limitations, and examples of all the discussed image restoration filters and the BID method are provided in this chapter. The measurements of image quality are detailed at the end of this chapter.

Chapter 3 provides a brief introduction of greatest common divisor (GCD) and its approximation AGCD. Then, the use of the 1D AGCD algorithm is presented in detail, including an introduction to the Sylvester resultant matrix, three pre-processing operations, the computation of the degree of AGCD, and the computation of the coefficients of AGCD.

Chapter 4 illustrates the developed algorithm of the BID for a separable PSF. The chapter starts by discussing the convolution operation of a blurred image and the fundamental mathematical theory of polynomial algebra for solving the BID problem. The implementation of the developed algorithm and some examples are provided at the end of this chapter.

Chapter 5 discusses in detail the developed algorithm of BID for a non-separable PSF. Some examples are provided at the end of this chapter. The problem that is faced during solving a non-separable PSF is demonstrated in the summary of this chapter and its solution is considered in the next chapter.

Chapter 6 discusses the modification of the developed algorithm of BID for a non-separable PSF. This algorithm is discussed in detail to examine the feasibility of using the DFT and non-linear structure matrix method APF in a totally BID for a non-separable PSF. Then, the implementation of this case of the developed algorithm and some examples are provided at the end of this chapter.

Chapter 7 discusses the performance of the developed method. Then, three different experiments are conducted to examine the performance of the developed method and other BID methods. The first experiment examines the BID performance with variable PSF width and additive noise. The second experiment examines whether the PSF size error affects the performance of BID methods. The third experiment examines the performance of the algebraic methods for the determination of the thus must go PSF. The fourth experiment performs face feature detection and circle feature detection on deblurred images resulting from various deblurring algorithms.

Chapter 8 provides a brief summary of the BID problem, as well as highlighting the most important details and the achievement of the developed algorithm. The suggestions for future research are presented at the end of this chapter.

1.9 Summary

Background information on the image restoration and BID problems have been introduced in this chapter along with an overview of the computational challenges of BID. Image restoration refers to a process that recovers the accurate and true image from its distorted version using limited or prior information about the degradation phenomenon. Given a blurred image and the blurring function, the original image is detected by solving an inverse problem. Computing the true image of a degraded image poses the most difficult problem for image processing owing to BID is an ill-posed problem.

The main objectives and contributions of this research have been provided in this chapter which is briefly that this research aim to develop a method for the solution of the problem of BID in which a blurred image is formed by spatially invariant and parametric blur. Under perfect boundary conditions, it is devised that an developed algorithm for BID problem, such that prior knowledge of the blurring function (PSF) and exact image are not required. The proposed algorithm is designed to incorporate the identification of PSF and the restoration of the image in two separate computational algorithms. The first algorithm computes the blur function from a blurred image, and is then used in the second algorithm to remove the blur from the blurred image. This will be done by representing the blurred and deblurred images, and the PSF as bivariate polynomials in which the coefficients are the pixel values. The PSF can be formulated as the approximate greatest common divisor (AGCD) of two degraded images that are represented by two bivariate polynomials. This research is carried out using approximate polynomial factorisation (APF) computational algorithms using a structure-preserving Sylvester resultant matrix method of two polynomials to compute a robust AGCD. The proposed work differs from other work in that it does not require any prior knowledge to recover an image from its blurred version.

In the end of this chapter, the structure of the remaining chapters of this thesis was detailed. In the next chapter, an introduction to the BID problem formulation and the fundamental information about the problem will be presented, along with highlighting the most used BID methods in the literature.

Chapter 2

Problem formulation

2.1 Introduction

Image restoration has been carried out through the use of blind image deconvolution (BID) over a period of four decades. A large number of image processing and mathematical techniques have been developed to tackle the BID problem. This chapter presents a brief introduction of some basic concepts in the field of BID, including the blurring model and different blur types. The final section discusses an overview of previous works in the image restoration field.

2.2 Concept of Image Deblurring

Image restoration refers to a process that recovers the accurate and true image from its distortions using limited or prior information of the blurring function. Image deblurring and image deconvolution are other expressions used to refer to image restoration. If it is assumed that the PSF is spatially invariant, then the blurring process can be modelled as a linear system, and the blurred image can be recorded as the output of the convolution of the original image (scene) with the PSF. Let $\mathcal{F}$ be the original image without any form of degradation, and $\mathcal{H}$ be the spatially invariant PSF. Then, the $\mathcal{G}$ be degraded form of the convolution of $\mathcal{H}$, and $\mathcal{F}$, and addition of noise $\mathcal{N}$. Mathematically, it is given by

$$\mathcal{G} = \mathcal{F} \otimes \mathcal{H} + \mathcal{N}, \tag{2.1}$$

where $\otimes$ denotes the 2-D convolution operator. Figure 2.1 demonstrates the blurring model of a camera.

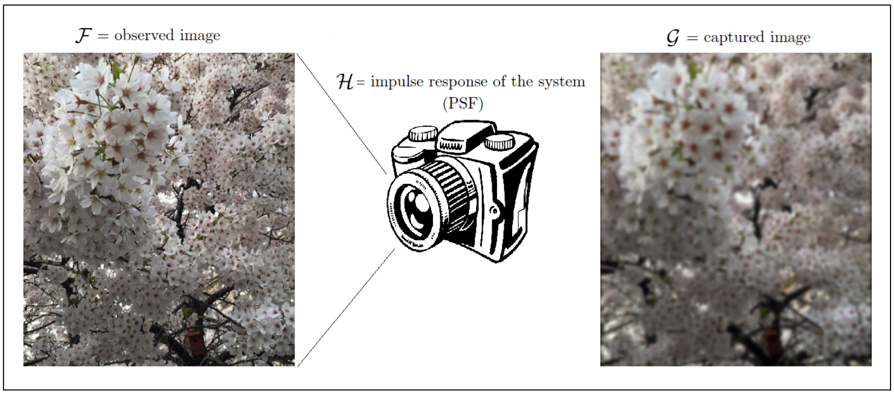

$\mathcal{F}$ = observed image
$\mathcal{H}$ = impulse response of the system (PSF)
$\mathcal{G}$ = captured image

Figure 2.1: Blurred image problem model of a camera.

As was mentioned in Chapter 1, the main goal of deblurring an image is to estimate the original image $\mathcal{F}$. In the absence of noise and when there is prior information of the $\mathcal{H}$, equation (2.1) can be used to resolve $\hat{\mathcal{F}}$, as an approximation of $\mathcal{F}$, by

$$\hat{\mathcal{F}} = \mathcal{H}^{-1}\mathcal{G}, \tag{2.2}$$

such that, $\hat{\mathcal{F}} \approx \mathcal{F}$.

This process is known as inverse filtering [16]. The original image can be estimated accurately if the exact $\mathcal{H}$ and components of convolution signal are known. However, in many cases, detailed knowledge about the original image and the blurring function are not available (see Figure 2.2); thus, the inverse filtering method is no longer feasible. Before trying to restore the blurred image, it is important to examine the types of degradation that occur in practice. Danelakis et al. [38] argue that there are five cases of blurred images depending on the artefacts' effect on the image (see Figure 2.3). The cases representing the degradation of the true image are listed as follows:

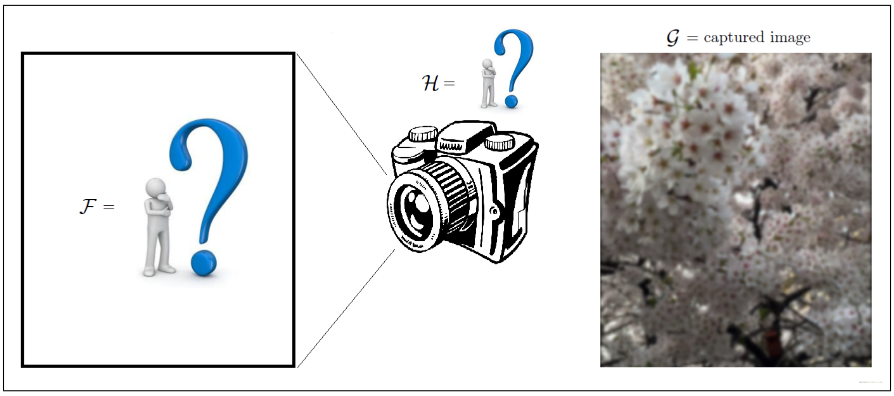


FIGURE 2.2: Real-life blind image deconvolution model.

- First case: The true image is only convolved by the blurring function. This process is commonly modelled as: $\mathcal{G} = \mathcal{F} \otimes \mathcal{H}$.
- Second case: The true image is blurred with additive noise N. This process is modelled as: $\mathcal{G} = \mathcal{F} + \mathcal{N}$. Thus $\mathcal{H} = \mathcal{I}$, the identity.
- Third case: The true image is convolved by the noisy blurring function. This process is modelled as: $\mathcal{G} = \mathcal{F} \otimes (\mathcal{H} + \mathcal{E})$ where $\mathcal{E}$ is the measurement error that has been added to the blurring function.
- Fourth case: The true image is convolved by the blurring function and noise is then added. This process is modelled as: $\mathcal{G} = \mathcal{F} \otimes \mathcal{H} + \mathcal{N}$.
- Fifth case: The true image is convolved by the noisy blurring function, and the additive noise is added. This process is modelled as: $\mathcal{G} = \mathcal{F} \otimes (\mathcal{H} + \mathcal{E}) + \mathcal{N}$.

The image restoration in the first case can be achieved by using deblurring filters for instance Wiener or regularisation filters, in cases where the PSF is known [38]. The image denoising algorithm [39, 40] can be used to solve the problem in the second case. On the other hand, the cases that are mostly found in real applications are those cases that combine PSF, noise and measurement error such as in the third, fourth and fifth cases. It should be taken into consideration that the terms $\mathcal{H}$, $\mathcal{E}$ and $\mathcal{N}$ are unknown. The literature shows various methods and mathematical tools that have been developed to resolve the BID, including iterative blind deconvolution algorithms [29, 41, 42], a constant modulus algorithm [43], a recursive filtering algorithm [34], maximum likelihood

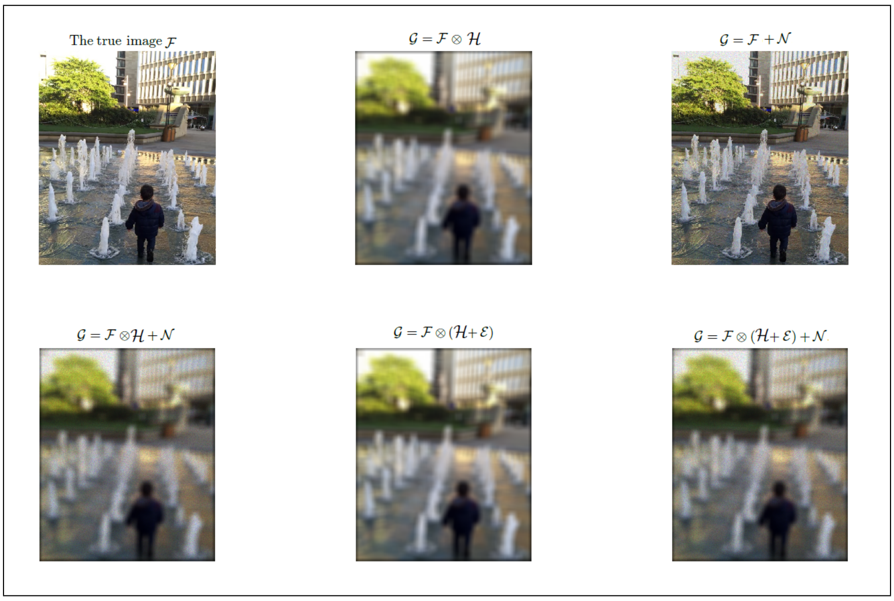

The true image $\mathcal{F}$
$\mathcal{G} = \mathcal{F} \otimes \mathcal{H}$
$\mathcal{G} = \mathcal{F} + \mathcal{N}$
$\mathcal{G} = \mathcal{F} \otimes \mathcal{H} + \mathcal{N}$
$\mathcal{G} = \mathcal{F} \otimes (\mathcal{H} + \mathcal{E})$
$\mathcal{G} = \mathcal{F} \otimes (\mathcal{H} + \mathcal{E}) + \mathcal{N}$

FIGURE 2.3: Five common cases of blurred images depending on the impact of artefacts on the image.

and projection [16, 32], total variation [31, 44], and greatest common divisor [1, 18, 19, 38].

This report focuses on examining the fifth category of blurred images, which is when a blurred image is formed by the convolution of the true image, and the noisy blurring function, and the small additive noise is then added. All of the blurred images that have been used in the experiment are from this category. In addition, important information that should be known about the PSF will be illustrated briefly in the next section.

2.3 The Point Spread Function

The serious distortion of the image comes from blurring. The point spread function (PSF) is a function that describes the procedure of blurring an image. Therefore, it is essential to understand the structure and the most important properties and characteristics of the PSF. This information helps to set up the precise mathematical model of the blurring image that is required for examining the image deblurring algorithm.

2.3.1 Definition and Properties

As has been mentioned above, the PSF is a function that describes the procedure of blurring an image, which is expressed in terms of the extent of the impact/influences of the pixel at position (i, j) in the exact image $\mathcal{F}$ by the pixel at position (s, t) in the blurred image $\mathcal{G}$. There are two types of blurring function, depending on its influences on the blurred image: spatially invariant blur and spatially variant blur.

With spatially invariant blur, the influence of the blur is the same for every pixel in the blurred image; see Figure 2.4. As can be seen from the figure, the images on the right hand side are blurred everywhere with the same intensity of blurring. The backgrounds and foregrounds of the images are affected equally by the blurring function and this is because the blurring function is invariant across the whole image. With a spatially variant blur, the blur varies over the image. Figure 2.5 is an example of a spatially variant blur. From the figure it can be seen that the objects in the background are much more blurred than the objects in the foreground. This is illustrated by contrasting the quality of the whiteboard, affected by blurring, in the background and the less blurrerd book in the foreground. The majority of existing BID methods in the literature were developed for tackling the spatially invariant PSF, and will also be examined in this research. Mathematically, the spatially invariant PSF is expressed as:

$$g(s,t) = \sum_{i=1}^{M} \sum_{j=1}^{N} f(i,j)h(s-i, t-j), \tag{2.3}$$

in which $g(s, t)$ and $f(i, j)$ are the pixel values at position (s, t) and (i, j) in $\mathcal{G}$ and $\mathcal{F}$, respectively, whereas M and N denote the size of the images $M \times N$. A shift invariant PSF is obtained by $h(s - i, t - j)$.

Equation (2.3) shows that the PSF is a linear operator. This means that each pixel value in $\mathcal{G}$ is a weighted linear combination of its pixel value and its neighbouring pixels. Additionally, the blurred image $\mathcal{G}$ may be obtained by the one-dimensional or two-dimensional convolution of the PSF and the exact image $\mathcal{F}$. However, in some cases

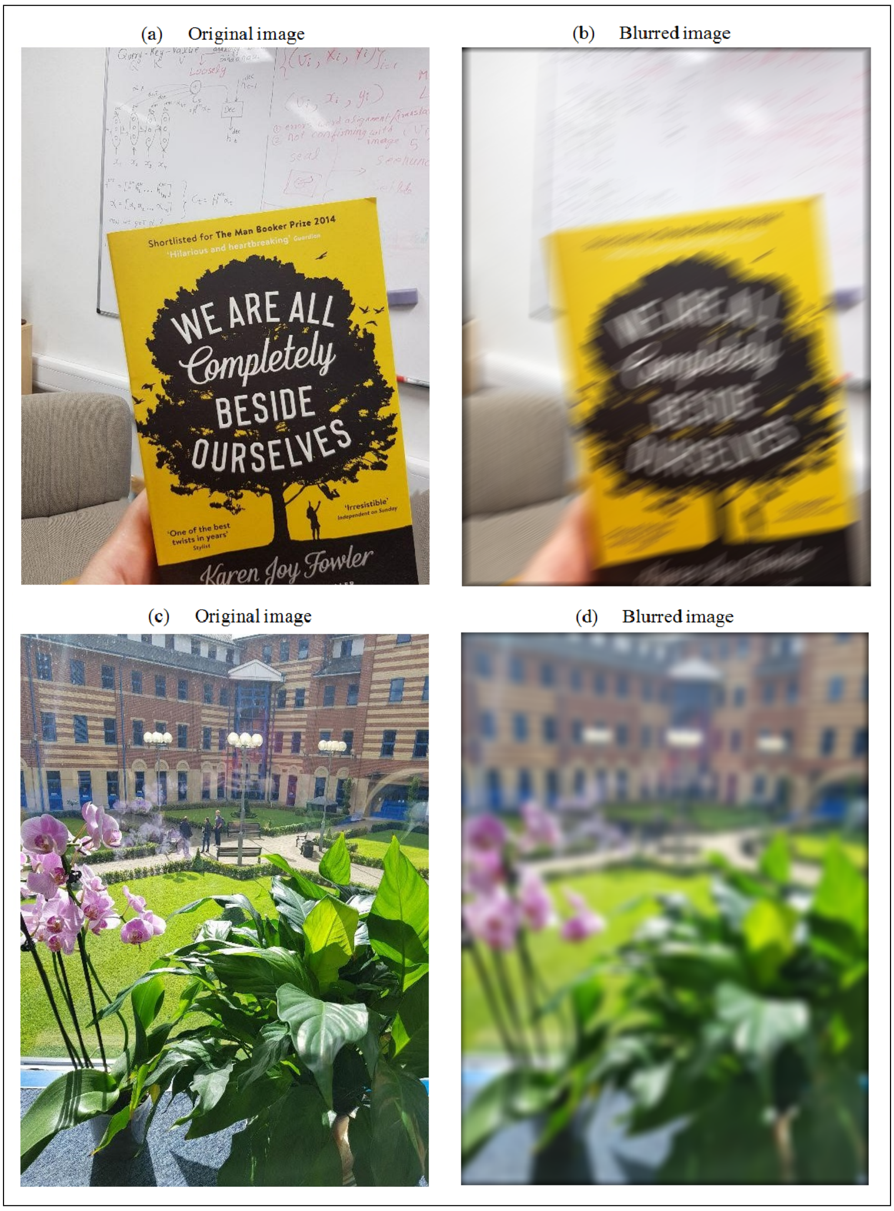


FIGURE 2.4: Spatially invariant blur.

in two-dimensional PSFs, the vertical and horizontal components of the PSF can be separated [10] and then the exact image can be affected independently by two one-dimensional PSFs. Mathematically, it is expressed as:

$$g(s,t) = \sum_{i=1}^{M} h_c(s-i) \sum_{j=1}^{N} f(i,j) h_r(t-j). \tag{2.4}$$

In this case, the PSF is called a separable PSF and its special structure is a rank-one matrix. The operation that can calculate the h_r and h_c is called a Kronecker product [10].

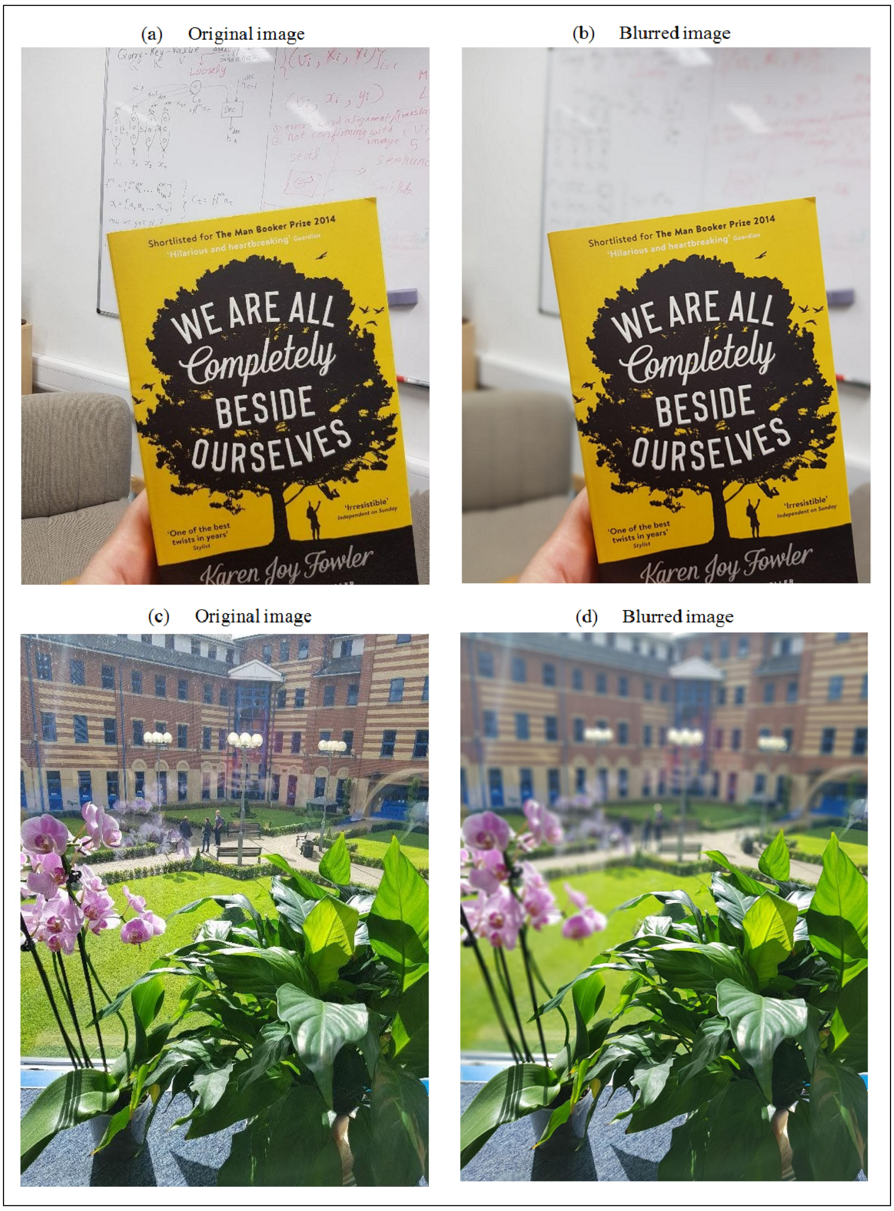


FIGURE 2.5: Spatially variant blur.

Most research in the literature compute separable, not non-separable PSF [45–48]. The separable PSF can be defined as

$$h(x, y) = h_c(y)h_r(x), \tag{2.5}$$

which mean the blurring function is decomposed into products of two functions that depend on the rows and columns of image, respectively. In other words, the rows and columns of blurring function can be computed independently. Whereas, the non-separable PSF rows and columns cannot be computed as a product of two functions, and it is expressed as

$$h(x, y) \neq h_c(y)h_r(x). \tag{2.6}$$

Kennedy and Samarasinghe [45] restore the true images corrupted by an unknown $2D$ PSF with a separable property using Constant Modulus Algorithm (CMA). They developed a new BID method for $2D$ separable PSF using algorithm based on archetype gradient descent that is used in $1D$ BID of communication systems. Shi and Reichenbach [46] investigate a cubic convolution for non-separable PSF in which the cubic convolution is a method for image interpolation. In the literature, most studies investigate the cubic convolution for $1D$ and $2D$ separable PSF. In 2003, Reichenbach and Geng [49] and, then in 2006, Shi and Reichenbach [46] derived a cubic convolution for $2D$ non-separable PSF.

Fang et al. [50] investigate a new BID of different perspective by introducing separable PSF to represent the inherent properties of the camera and scene system. Lee and Hwang [47, 48] present a BID method for mixture of a separable Gaussian Blur. They assume that the PSF is a separable circular convolution and the $2D$ PSF derivative to $1D$ PSF using Kronecker product.

The PSF has the following properties:

- The sum of the component values of the PSF matrix equals one, because it is assumed the captured image brightness should remain unchanged after deconvolution.
- The size of the PSF matrix is much smaller than the size of the exact image $\mathcal{F}$.

- Blurring is a local phenomenon, which means that the PSF is restricted to a small area about its centre and the distance from its centre will be a zero beyond.
- The PSF is spatially invariant. This is assumed in this research study but it may not be satisfied in practice.

To conclude, the PSF will contain all information about the blurring of the exact image $\mathcal{F}$ [10].

PSFs can be commonly classified into two groups based on their shape/form: parametric and non-parametric. Parametric PSFs can be easily described using a functional or parametric form. Specifically, parametric PSFs can be generated by an equation. In contrast, non-parametric PSFs (arbitrarily shaped) usually cannot be generated by an equation of their parameters because they have a complex shape. In the context of this research, PSF refers to the spatially invariant and parametric type of blurring function. The following shows the most common type of PSFs.

2.3.2 Most Common PSFs

The blurring in images arises from many sources during the capture process. For example, it can arise from limitations of the optical system, camera and object motion, defocused objects, lens error, and environmental effects such as atmospheric perturbations. The most common types of blur are Gaussian blur, motion blur and out-of-focus blur, which are shown in Figure 2.6:

- **Gaussian blur** of this kind is normally caused by lens errors or atmospheric perturbations. The Gaussian filter in 2-D over PSF rows and columns, i and j, and its centre at point (k, l), according to [10] is given as

$$h(i,j) = \exp\left(-\frac{1}{2}[i-k \qquad j-l]\begin{bmatrix} s_1^2 & \sigma^2 \\ \sigma^2 & s_2^2 \end{bmatrix}^{-1}\begin{bmatrix} i-k \\ j-l \end{bmatrix}\right), \tag{2.7}$$

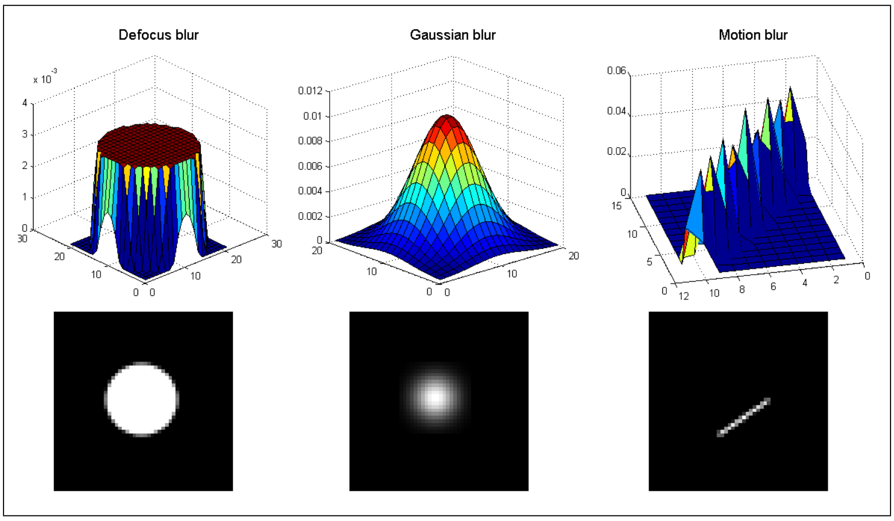


FIGURE 2.6: Different kinds of PSF

where

$$\sum_{i,j} h(i,j) = 1, \tag{2.8}$$

in which s_1, s_2 and σ are parameters that control the width and orientation of the PSF. The shape of Gaussian blurs display exponential decay from the centre of the image. Also, 2-D Gaussian blurs will be separable (rotationally symmetric), if $s_1 = s_2$.

- **Motion blur** occurs when the camera or object motion is in the x-direction during the image capture time. According to [11], the motion PSF is given by (2.9), where L denotes the length of motion and the angle is denoted by φ.

$$h(i,j;L,\varphi) = \left\{ \begin{array}{llll} \frac{1}{L} & if & \sqrt{i^2+j^2} \leqslant \frac{L}{2} \quad and & \frac{i}{j} = -\tan\varphi \\ 0 & & elsewhere & \end{array} \right\}, \tag{2.9}$$

where i and j are the PSF pixel coordinates.

- **Defocus blur** comes from outside the focus of the camera. If the camera system has a circular aperture, the captured image of any point source is a uniform disk with radius R. Therefore, the out-of-focus PSF, that coordinates i and j, and its centre (k, l), is given by (2.10)

$$h(i, j; R) = \left\{ \begin{array}{lll} \frac{1}{\pi R^2} & if & \sqrt{(i-k)^2 + (j-l)^2} \leqslant R^2 \\ 0 & & elsewhere \end{array} \right\}, \tag{2.10}$$

Figure 2.7 illustrates what an exact image affected by such blurs looks like.

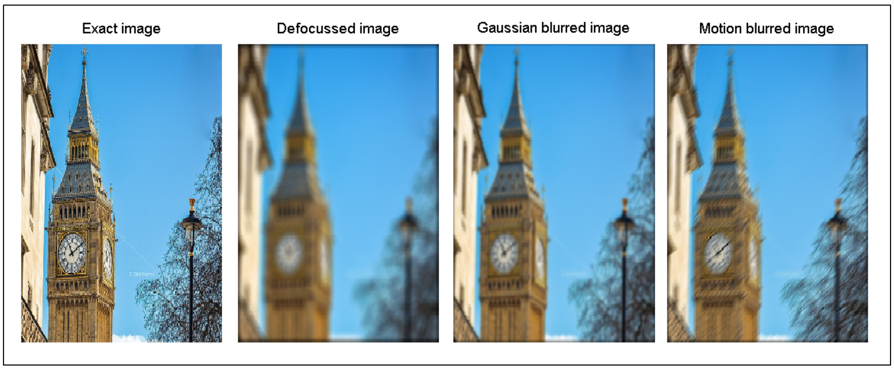


Figure 2.7: Examples of blurred images

2.4 Image Restoration Filters

This section reviews some of the classical image restoration methods that have been studied in this research. There are several BID methods used to restore blurred images, requiring advanced knowledge about the problem. BID is classified into two main approaches [16] :

- One procedure is to identify the PSF for the true image separately to use it later on with the known classical image restoration methods. Estimating the PSF and the true image is a separated procedure.
- The other procedure is to incorporate the process of identifying PSFs with the restoration algorithm. This involves merging both the estimation and the PSF true image simultaneously; this results in the development of more complicated algorithms.

Some of the most frequently used methods in BID problems are discussed in the following sections.

2.4.1 Inverse Filtering

The ideal of the inverse filter is to deblur the distorted image by estimating the inverse of the PSF that blurred the image, and then apply it to the distorted image to recover the original image. In the case where there is an absence of noise, it can be easy to directly apply the inverse filter to the distorted image in the spectral (frequency) domain [11], since the process of convolution will be reformed into multiplication. The inverse filtering process is given by

$$\hat{\mathcal{F}} = \mathcal{H}^{-1}\mathcal{G}, \tag{2.11}$$

where $\hat{\mathcal{F}}$ is an estimation of the exact image $\mathcal{F}$. As mentioned above, in most cases of real blurred images, the PSF is not available. On the other hand, the PSF may be approximated in some situations. An example is when the image had been blurred due to linear translation or movement of the image pixels during the image capture. In this case, the PSF can be represented in the spectral domain by a sine function [11]. Thus, in the frequency domain, one can multiply the blurred image with the inverse of the PSF to estimate the original image. However, the inverse filter is still a simple method that cannot be used in practical cases of deblurring. This is due to the difficulties of estimating the correct values or coefficients of the corresponding PSF. These difficulties of estimating the correct values of PSF are due to the presence of noise amplification during deblurring and frequency domain zeros. These issues limit the inverse filter from being utilised in many image restoration applications.

2.4.2 Wiener Filtering

Inverse filtering reduces a single case of degradation at a time because it is a simple approach [11], and it is very sensitive to additive noise. Therefore, a restoration algorithm has to be developed for each type of degradation. Wiener filtering is used for the linear estimation of the original image. The approach is based on a least squares method. In the Fourier domain, the Wiener filter can be expressed as [8],

$$\hat{\mathcal{F}} = \left[\left(\frac{\mathcal{H}^*}{|\mathcal{H}|^2 + \delta}\right)\right] \qquad where \quad \delta = \frac{|\mathcal{G}|^2}{|\mathcal{F}|^2}. \tag{2.12}$$

This filter is described in the frequency domain and requires statistical information about the exact image $\mathcal{F}$ and noise $\mathcal{N}$. It removes and inverts blurring and the additive noise, using the Mean Squared Error (MSE) criterion [8]. In other words, it determines the blurring function that smoothes noise by minimising the overall MSE. Figure 2.8, shows the image restoration output using a Wiener filter. The Wiener filter assumes the exact PSF and noise-to-signal power ratio (PNSR) are known and must be included in its function calls. An estimate of the PNSR is

$$PNSR = \frac{\|\mathcal{G} - \mathcal{F}\|_F^2}{\|F\|_F^2}, \tag{2.13}$$

where the subscript F denotes the Frobenius norm.

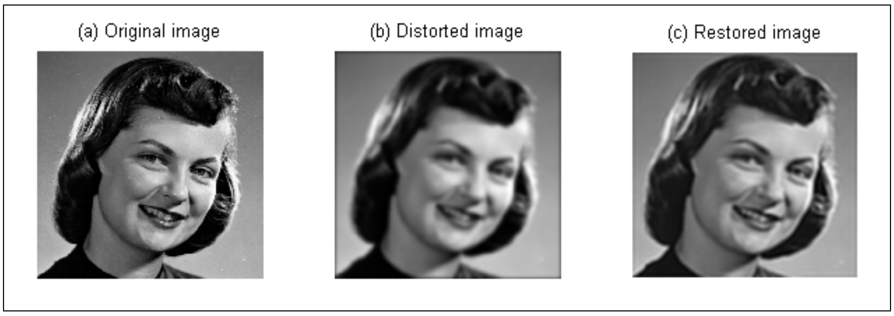


FIGURE 2.8: Image de-blurring using Wiener filter. (a) an clear image; (b) a blurred image; and (c) the restored image.

2.4.3 Iterative Blind Deconvolution Method

The iterative blind deconvolution method (IBD) used to deblur the degraded image use the Fast Fourier Transform (FFT) and deterministic constraints in the form of finite support and non-negativity constraints. Many degraded images in the spatial domain have been restored by convolution methods, such as IBD method, in the spectral domain when a given image is too large. These methods are also used when prior information cannot be represented appropriately in the spectral domain such as the positivity of image intensities [51].

Let $\hat{F}, G$ and $\hat{H}$ represent the Fourier transforms of the corresponding signals of the linearly estimated image $\hat{f}$, the degraded image g, and the estimated PSF $\hat{h}$, consecutively. The iterative algorithm is shown in Figure 2.9, as given by [16, 34], where the subscript itr denotes the number of algorithm iterations.

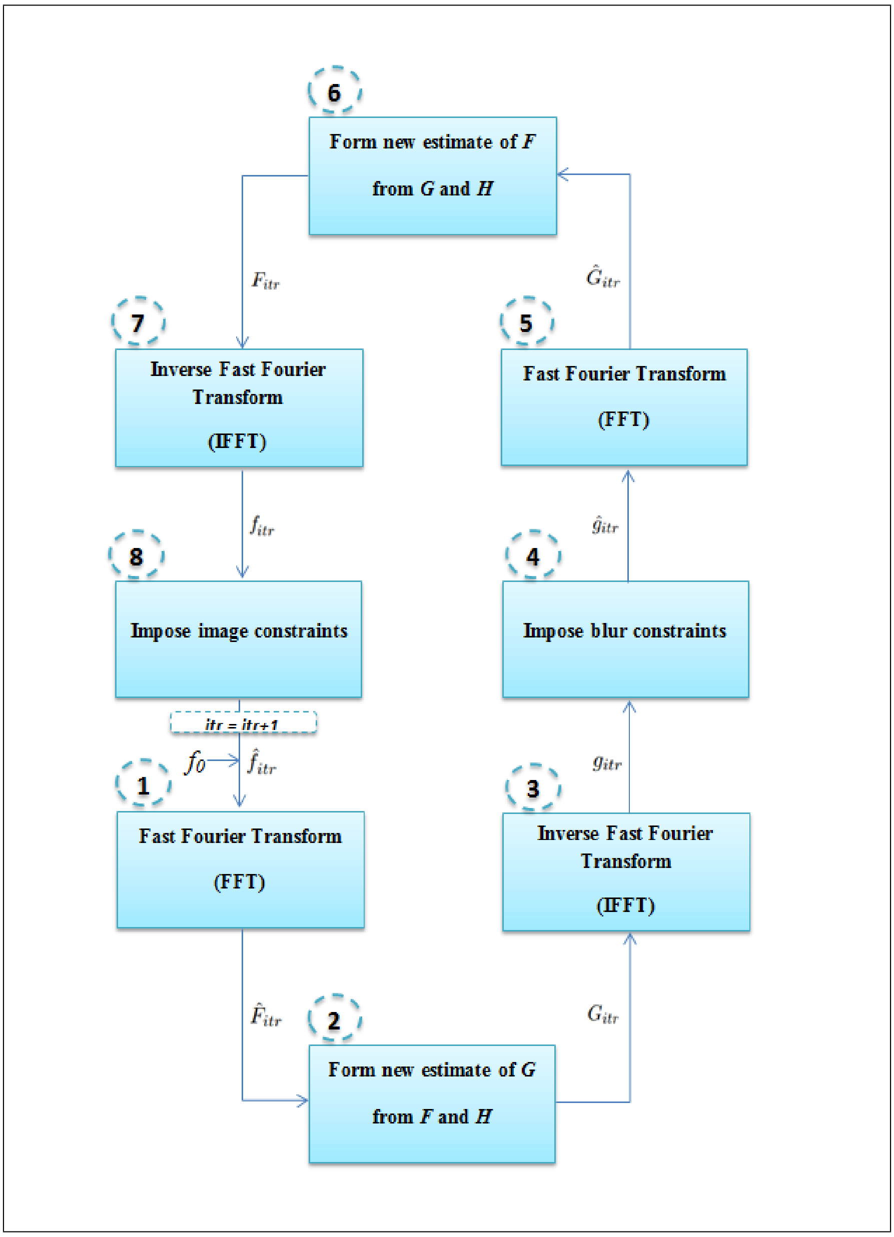

6
Form new estimate of F from G and H
F_{itr}
$\hat{G}_{itr}$
7
Inverse Fast Fourier Transform (IFFT)
5
Fast Fourier Transform (FFT)
f_{itr}
$\hat{g}_{itr}$
8
Impose image constraints
4
Impose blur constraints
$itr = itr+1$
f_0
$\hat{f}_{itr}$
1
Fast Fourier Transform (FFT)
3
Inverse Fast Fourier Transform (IFFT)
g_{itr}
$\hat{F}_{itr}$
2
Form new estimate of G from F and H
G_{itr}

FIGURE 2.9: Iterative blind deconvolution method.

The process of the iterative blind deconvolution method, which is shown in Figure 2.9, can be summarised as follows:

1. The input to the process is the initial estimation f_0 and it is a non-negative value.
2. The Fast Fourier Transform is applied to give $\hat{F}_{itr}$, which is then inverted by using the inverse filter. Then, it is used to form a new estimate of G, $\hat{G}_{itr}$.
3. The Inverse Fast Fourier Transform (IFFT) is applied to $\hat{G}_{itr}$ to transform to g_{itr}.
4. Then, the image non-negativity constraints are imposed to reveal a positive constrained estimation of g_{itr}.
5. After that, the spectrum of $\hat{G}_{itr}$ is provided by a Fourier Transform of g_{itr}.
6. Then, $\hat{F}_{itr}$ is estimated by inverting $\hat{G}_{itr}$ to form an inverse filter, multiplied by $\hat{H}_{itr}$.
7. The Inverse Fourier Transform is applied to $\hat{F}_{itr}$ to give f_{itr}.
8. Finally, the image constraints, being non-negative, are applied to estimate $\hat{f}_{itr}$.

These steps describe a single iteration of the iterative method, but its loop is repeated until it finds, g, two positive functions with the required convolution.

Unfortunately, the IBD algorithm has two main problems:

- It is difficult to define the inverse filter in some regions with small values of the inverted function.
- Both of the frequencies $\hat{F}_{itr}$ and $\hat{G}_{itr}$ have spectral zeros that provide no knowledge about that spatial frequency being a part of the blurring process.

The IBD method has low complexity, so it has popular usage [16, 34]. However, the implementation of this method differs depending on the original image and the PSF. The advantage of this method is the robustness of the IBD method to noise [11, 52] compared to the previously discussed BID methods. The disadvantage of this method is that the IBD method suffers from instability, convergence, uncertain uniqueness, and sensitivity to the initial image estimate.

2.4.4 Richardson-Lucy Algorithm

The Richardson-Lucy method is one of the iterative blind deconvolution methods [6] that is based on Bayes' theorem of conditional probability. This theorem considers the original image F, PSF H, and blurred image G as probability functions. Thus, Bayes' theorem is represented as follows

$$P(F|G_{itr}) = \frac{P(G_{itr}|F)P(F)}{\sum\limits_{i,j=1}^{M,N} P(G_{itr}|F)P(F)}, \tag{2.14}$$

where the subscript itr denotes the number of algorithm iterations and i, j indicate the $(i, j)^{th}$ pixel in the image of size $M \times N$. And G_{itr} is considered based on its dependence on F by

$$P(F) = \sum_{i,j=1}^{M,N} P(F|G_{itr})P(G_{itr}), \tag{2.15}$$

and

$$P(F|G_{itr}) = P(F|G_{itr})/P(G_{itr}). \tag{2.16}$$

The substitution of (2.14) into (2.15) gives the following

$$P(F) = P(F)\sum_{itr} \frac{P(G_{itr}|F)P(G_{itr})}{\sum\limits_{i,j=1}^{M,N} P(G_{itr}|F)P(F)}, \tag{2.17}$$

It is also recommended for the $P(F)$ term on the right-hand side, and its initial estimation to be determined using (2.14), which results in the IBD method given in (2.18)

$$P_{itr+1}(F) = P_{itr}(F) \sum_{itr} \frac{P(G_{itr}|F)P(G_{itr})}{\sum\limits_{i,j=1}^{N,M} P(G_{itr}|F)P(F)}. \tag{2.18}$$

The disadvantage of this method is that it requires an initial assumption of the support size of the PSF. The support size of the PSF must be either estimated or known. Consequently, the Richardson-Lucy algorithm is non-blind.

Figure 2.10 shows the image restoration output using the Richardson-Lucy algorithm. The Richardson-Lucy algorithm assumes the exact PSF is known and the number of iterations must be specified.

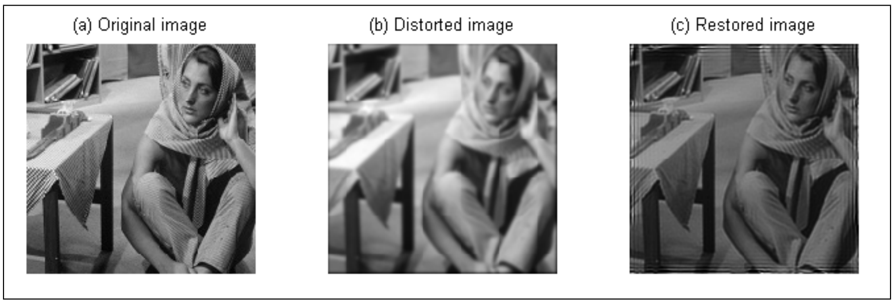


Figure 2.10: Image de-blurring using Richardson-Lucy algorithm. (a) a clear image; (b) a blurred image; and (c) the restored image.

2.4.5 Regularisation-Based Deblurring Algorithm

As was pointed out earlier, the image estimation of the blurring convolution model, that is present in (2.1) through inverse filtering, is given as follows:

$$\hat{F} = \frac{G}{H} = F + \frac{N}{H}, \tag{2.19}$$

The BID is an ill-posed inverse problem because a small bounded error in the input leads to an amplification of the high frequency error in the output [9]. Therefore, in the presence of noise, the filter defined in (2.19) behaves poorly and it is better to use the least squares method or regularisation methods.

The theory of regularisation methods can reduce the effect of noise in the restored image by incorporating prior information about the noise or original data. Consequently, it can achieve meaningful solutions to the ill-posed problem. Let L represent a regularisation operator in the frequency domain. In order to determine the constrained regularisation L, there are two methods: truncated singular value decomposition (TSVD) and Tikhonov regularisation [10, 53]. Figure 2.11, shows the image restoration result using Regularisation-Based Deblurring method. The Regularisation filter assumed the exact PSF and the noise power (NP) are known. An estimate of the NP is given by

$$NP = \|\mathcal{G} - \mathcal{F}\|_F^2, \tag{2.20}$$

where the subscript F denotes the Frobenius norm.

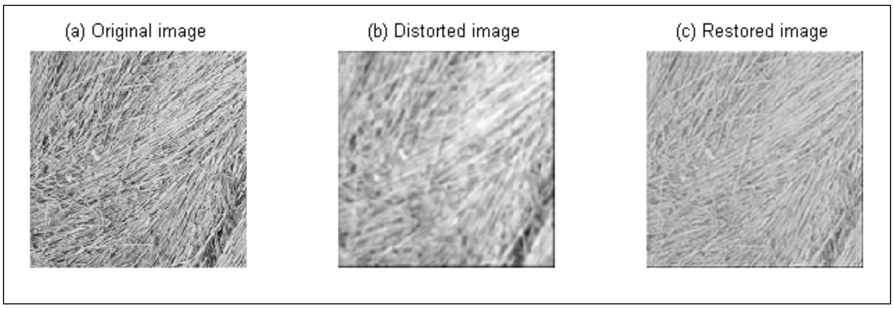


FIGURE 2.11: Image de-blurring using Regularisation-Based Deblurring filter. (a) an clear image; (b) a blurred image; and (c) the restored image.

2.4.6 Maximum Likelihood Restoration Method

Maximum Likelihood Restoration (ML) is an enhanced technique of the iterative constrained algorithm. It estimates iteratively the optimisation of $\hat{F}$ and $\hat{H}$ based on a statistical model. ML is similar to most of the image deblurring methods in that ML requires prior information of the exact image and the PSF. More precisely, the $2D$ autoregressive (AR) model of the exact image $f(x, y)$ is given as follows

$$f(x, y) = a_{01}f(x, y-1) + a_{11}f(x-1, y-1) + a_{10}f(x-1, y) + \nu(x, y), \tag{2.21}$$

where $\nu(x, y)$ is an unpredictable spatial component which is the modeling error and a_{ij} are the AR coefficients [7]. ML attempts to estimate the AR model coefficients of the exact image, variance of additive noise $n(x, y)$ and the PSF in order to derive restoration filters. Thus, it is assumed that the parameters a_{ij}, $\nu(x, y)$, and the white noise equal to variance σ_ν are unknown. In this case, the Maximum-Likelihood method (ML) is used to define the parameter set, which is $\theta = \{a_{ij}, \sigma_\nu^2, \sigma_n^2, H\}$, from the given distorted image. In which σ_ν^2 and σ_n^2 are the variances of $\nu(x, y)$ and additive noise $n(x, y)$, respectively.

This gives the best estimation for the parameter set of highest probability. This can be formulated for Gaussian distribution by [7]

$$L_\theta = \sum_{u,v} \left(\log P(u, v) + \frac{g(u, v)g^*(u, v)}{P(u, v)} \right), \tag{2.22}$$

in which $P(u, v)$ is the probability density function (PDF) of the blurred image. The PDF is obtained by combining the AR model in (2.21) and the image deblurring model in (2.1), leading to

$$P(u, v)|_{g,\theta} = \sigma_\nu^2 \frac{|H(u, v)|^2}{|1 - A(u, v)|^2} + \sigma_n^2, \tag{2.23}$$

where $H(u, v)$ is the PSF and $A(u, v)$ is the coefficient a_{ij} in a 2D discrete Fourier Transform. Thus, maximising the log likelihood function and estimating the optimal parameters of θ, the image restoration will then be achieved [7]. However, there are limitations of this method: the main one is that in order to achieve an acceptable estimation of PSF, more constraints should be applied; for example, the sum of PSF is equal to 1. Moreover, the log likelihood function is a non-linear optimisation method and there are no unique solutions available [54]. Figure 2.12 shows the image restoration result using the Maximum Likelihood Restoration Method. The ML method requires an initial estimate of the PSF and returns an improved estimate of the PSF and a deblurred image.

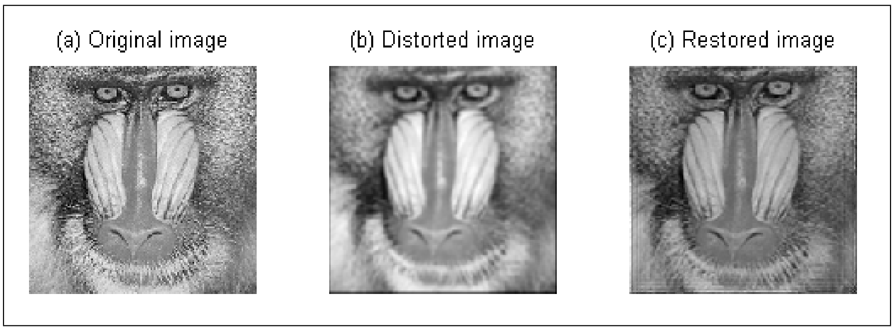


FIGURE 2.12: Image deblurring using the Maximum Likelihood Restoration Method. (a) a clear image; (b) a blurred image; (c) the restored image.

2.4.7 Total Variation Blind Deconvolution

The Total Variation Blind Image Deconvolution (TVBD) method is a regularization-based deblurring algorithms. Total Variation Blind Image Deconvolution was developed in 1998 by Chan and Wong [55]. From that time until the present day, various research studies have been conducted to enhance and improve this method. The proposed method will be compared with the most recent TVBD method put forward in 2014 by Perron and Favaro [3]. Briefly, the TVBD method can be introduced as follows. Suppose that the model blur degradation is

$$g = f \otimes h + n \tag{2.24}$$

where g is a blurred image, f the original image, h the blur kernel, and n noise. Then, to solve the BID problem a classic regularized minimization approach could be used

$$\min_{f,h} \|fh - g\|_2^2 + \lambda J(f) + \gamma W(h), \tag{2.25}$$

where $\lambda J(f)$ and $\gamma W(h)$ are regularization terms, in which λ and γ are non-negative regularization parameters that weigh their contribution and the functional $J(f)$ and $W(h)$ are the smoothness priors for f and h, respectively. To enhance the convergence of regularization algorithms, Chan and Wong [55] consider additional constraints which is positivity of h entries and integration to 1

$$\|h\|_1 \doteq \int |h(\mathbf{x})| d\mathbf{x} = 1, \qquad h(\mathbf{x}) \geq 0, \qquad f(\mathbf{x}) \geq 0, \tag{2.26}$$

where with x indicates either 1D or 2D coordinates. Furthermore, they do not use any regularization on the h by setting $\gamma = 0$. Therefore, by combining the constraints in (2.26) with (2.25) the following minimization is studied:

$$\min_{f,h} \|fh - g\|_2^2 + \lambda J(f) \qquad \text{subject to } h \geq 0, \qquad \|h\|_1 = 1 \tag{2.27}$$

Chan and Wong used a sparse gradient prior via total variation $J(f) = \|f\|_{BV} \doteq \int \|\nabla f(\mathbf{x})\|_2 d\mathbf{x}$ or $J(f) = \|f_x\|_1 + \|f_y\|_1$, with $\nabla f \doteq [f_x \quad f_y]^T$ as the gradient of f and $\mathbf{x} \doteq [x \quad y]^T$. Then, they used the alternating minimization algorithm (AM) to solve the problem in (2.27). The algorithm alternates between the estimation of the original image f given the latest estimate of blur kernel h, and the estimation of the blur kernel h given the updated image f. Perron and Favaro [3] enhanced the iterative algorithm of AM to be Projected Alternating Minimization (PAM). Perron and Favaro [3] made the imposing constraints as something that was selected sequentially rather than during the gradient descent on h, and it was seen as rather acceptable of the correct procedure AM. The pseudo-code of their algorithm is summarized in Algorithm 2.1:

Algorithm 2.1: Total Variation Blind Image Deconvolution (TVBID)

Input: g, size of blur, initial large λ, final λ_{min}

Output: f, h

```
$f_0 \leftarrow pad(g)$;
$h_0 \leftarrow uniform$;
while not converged do
    $f^{t+1} \leftarrow f^t - \varepsilon_f(h^t \bullet (h^t \circ f^t - g) - \lambda \nabla \cdot \frac{\nabla f^t}{|\nabla f^t|})$;
    $h^{t+1/3} \leftarrow h^t - \varepsilon_h(f^{t+1} \circ (h^t \circ f^{t+1} - g))$;
```

```
        $h^{t+2/3} \leftarrow \max\{h^{t+1/3}, 0\}$;
        $h^{t+1} \leftarrow \frac{h^{t+2/3}}{h^{\|t+2/3\|_1}}$;
        $\lambda \leftarrow \max\{0.99\lambda, \lambda_{min}\}$;
        $t \leftarrow t+1$;
    end
    $f \leftarrow f^{t+1}$;
    $h \leftarrow h^{t+1}$;
```

Figure 2.13 shows the image restoration result using the Total Variation Blind Deconvolution Method that was enhanced by Perron and Favaro [3].

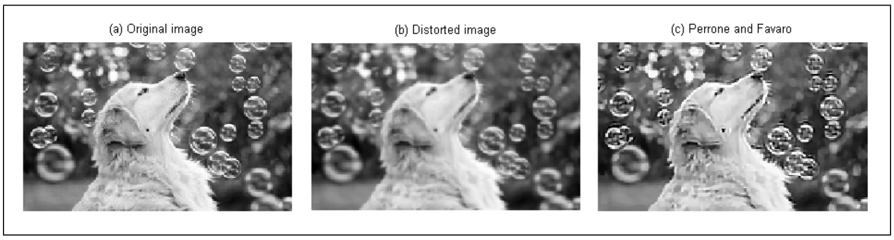


FIGURE 2.13: Image deblurring using Total Variation Blind Deconvolution [3]. (a) a clear image; (b) a blurred image; (c) the restored image.

2.4.8 Maximum a Posteriori (MAP)

The maximum a posteriori (MAP) distribution is another BID approach. This method seeks a $(\hat{f}, \hat{h})$ pair, that maximizes a posteriori probability:

$$(\hat{f}, \hat{h}) = \arg\max \log p(f, h|g). \tag{2.28}$$

The $\text{MAP}_{f,h}$ should minimize the following objective [56]

$$p(f, h|g) \propto p(g|f, h)p(f)p(h), \tag{2.29}$$

where $p(f)$ and $p(h)$ are the image prior and kernel prior, respectively. The likelihood term $p(g|f,h)$ is a data fitting term and is equal to

$$\log p(g|f,h) = -\lambda \|h \otimes f - g\|^2. \tag{2.30}$$

The assumption of a uniform prior on h and the prior $p(f)$ favours natural images where their gradient distribution is sparse. It can be measured as [56]

$$\log p(f) = -\sum_i |j_{x,i}(f)|^\alpha + |j_{y,i}(f)|^\alpha + C, \tag{2.31}$$

where $j_{x,i}(f)$ and $j_{y,i}(f)$ are the horizontal and vertical derivations at pixel i. The simplest non-trivial filters are $[1,-1]$ and $[1,-1]^T$ [4, 56]. Moreover, C is a constant normalization term. Furthermore, the exponent value α is a sparse prior and it has various options including the Gaussian prior $\alpha = 2$ and the Laplacian prior $\alpha = 1$; natural images often correspond to α in the range of $[0.5, 0.8]$ [56]. Most BID methods in the literature are MAP estimators [4], but, actually they do not find a global minimizer because this is intractable and may sometime be counterproductive [4].

The important and general idea of the highlighted methods has been illustrated in the following studies. Firstly, Fergus et al. (2006) show that the alternating minimization approach, which means computing, simultaneously, the exact image (f) and the kernel (h), fails to compute $\text{MAP}_{f,h}$ correctly. Therefore, instead they used MAP_h approach [23], which computes the kernel (h) by maximizing the marginalized distribution. Then the exact image (f) was estimated by deblurring the blurred image with estimated kernel h. For priors, a variational Bayesian approach and a Gaussian mixture model have been used in this method. On the other hand, Shan et al. (2008) [57] use an iterative minimization approach $(\text{MAP}_{f,h})$. In this approach a kernel prior and exact image prior are used. The blur kernel is modelled by an exponentially distributed prior. The exact image prior is designed to achieve two objectives: A prior to reduce ill-posedness of the deconvolution problem and also a prior to reduce ringing artifacts in the deblurred images. Therefore this prior was composed from two priors: local prior and global prior.

Cho and Lee (2009) [58] modified the alternating minimization (MAP$_{f,h}$) approach to improve the observed sharp edge restoration and noise suppression in smooth regions. Therefore, estimation of the kernel h can yield good results at the edges. They used a shock filter to predict and restore strong edges. A shock filter is a tool that enhances the image features. So, the sharp edges can be recovered from distorted step signals. Later on, Levin et al. (2011) [59] used MAP$_h$ algorithm instead, based on Fergus et al. (2006) [23]. Levin et al. derived a simple approximation of the MAP$_h$ algorithm and it is a modification of MAP$_{f,h}$ algorithms. In this, algorithm priors are used on images and they show that good results are obtained using Gaussian priors. However sparse priors improve performance but they have no closed-form formula. They suggest the use of an approximation model based on samples to approximate these formulae. For natural images Levin et al. suggest the use of prior values in the range [0.6,0.8].

In 2012, Babacan et al. [5] proposed a general framework based on Bayesian inference with Super Gaussian priors. In this approach a large family of priors are used to model natural images and it includes some of the existing modeling and inference methods as special cases. The author states that this framework is very effective, efficient and flexible. In 2013, Xu et al. [60] and Shearer et al. [4] used and developed the alternating minimization approach MAP$_{f,h}$. Xu et al. [60] introduced a new sparse L_0 approximation scheme on the image. In this approach, a shock filter is not used, but a cost L_0 function is used instead. This cost function is approximated by a family of loss functions incorporated in a new regularization term. This regularization term is incorporated in the optimization objective and leads to consistent minimization and fast convergence. Shearer et al. [4] used incremental sparse edge approximation for the blur kernels. This method is developed for camera shake blurred images and starts by estimating a blur kernel from the strongest edges in the image and then refined gradually by allowing weaker and weaker image edges. Finally, in 2015 Zhou et al. [61] used a Dirichlet distribution to approximate the MAP distribution of blur. They proposed a variational Dirichlet approximation (VD) that can also be applied with MAP BID methods whose blur estimation models are quadratic. For example, Levin et al. [59] and Babacan et al. [5] state that the algorithms of MAP distribution differ primarily in the assumption and the prior information about the blurred image they include to perform the deconvolution task. Sometimes, MAP implementations work well but other times they fail.

It is observed that the MAP method suffers from a no-blur solution. No-blur solution occurs when the calculated kernel is a scalar value and the deblurred image is identically equal to the blurred image, which is one of the undesirable solutions.

The developed method result was compared with Babacan et al. [5] and Shearer et al. [4] in Chapter 5, Figure 2.14 shows an example of their results. These algorithms are suitable for small PSF functions, but, the quality of the deblurred images depends on the prior. The estimation of the prior values of the original image and the kernel is not an easy task and different authors use different methods for their estimation.

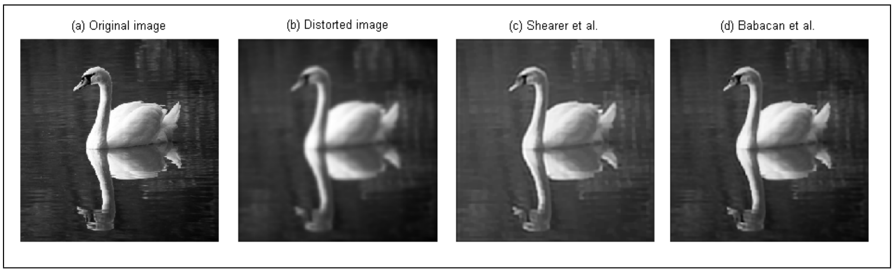


FIGURE 2.14: Image deblurring using the MAP-based method. (a) a clear image; (b) a blurred image; (c) the restored image based on implementing the method of Shearer et al. [4]; (d) the restored image based on implementing the method of Babacan et al. [5].

2.4.9 Nongaussianity Approach to Blind Image Deconvolution

Nongaussianity is one of the central concepts in the theory of Independent Component Analysis (ICA) [62–66]. It is measured through the use of higher order statistics. ICA is a data analysis method that is widely used for blind source separation for one dimensional data and recently has been applied to two dimensional data such as images [67–69]. In ICA, the classical and most widely used statistics for measuring nongaussianity are Kurtosis and Negentropy [63–65, 70]. These statistics are used to measure the closeness of a random variable distribution to that of a Gaussian random variable distribution.

The Central Limit Theorem states that the distribution of the sum of independent random variables tends towards that of a Gaussian distribution as the number of the summed random variables increases. In other words the distribution of two random variables is closer to Gaussian distribution than any of the summed random variables distributions.

Kurtosis is a fourth-order cumulant statistic that measures how high or low the peak of the random variables distribution is. The normalised version of this measure is defined as follows [63, 66]:

$$Kurt(y) = E[y^4] - 3(E[y^2])^2. \tag{2.32}$$

For a normalised random variable, the kurtosis simplifies to $Kurt(y) = E[y^4] - 3$. This implies that the Kurtosis of a normalised Gaussian random variable is equal to zero. It follows that kurtosis can have positive values in which case the random variable distribution has higher peak than the Gaussian random variable and also it can have negative values where the distribution peak is flatter and lower than that of a Gaussian random variable. Nongaussianity of a random variable is measured by the absolute value of the kurtosis.

$$nongaussianity(y) = abs(E[y^4] - 3(E[y^2])^2). \tag{2.33}$$

As the nongaussianity value of a random variable increases, the random variable distribution becomes farther away from that of a Gaussian random variable distribution. Nongaussianity as defined with the above formula becomes zero for standardised Gaussian random variables and greater than zero for other standardised nongaussian variables. The second widely used measure of nongaussianity is the Negentropy. This statistic captures the idea that the more random and unstructured the variable is, the larger is its entropy. Nogaussianity is measured by subtracting the random variable entropy from that of a Gaussian random variable with the same covariance (which has the largest entropy value). Negentropy is also approximated by the following formula:

$$nongaussianity(y) = \frac{1}{2}E[y^3]^2 + \frac{1}{48}Kurt(y)^2. \tag{2.34}$$

The Kurtosis and Negentropy approximation are two classical measures of nongaussianity. Given that the kurtosis is sensitive to outliers, the above negentropy approximation is also sensitive to outliers. Hence the above definitions are not robust enough in application; more robust version of these measures has been developed, see for example [63, 66, 67, 70].

Independent Component Analysis depends greatly on the nongaussianity concept. The basic data model of the ICA theory is as follows:

$$X = As. \tag{2.35}$$

Where X is the random vector variable whose components are the mixtures $X_1, \ldots, X_n$, s is the sources random vector variable with components $S_1, \ldots, S_n$ and A is the mixing matrix with components a_{ij}. In this model the independent components $S_1, \ldots, S_n$ are latent variables (unknown) and the mixing matrix A is assumed to be unknown. The only data available is the observed (measured) $X_1, \ldots, X_n$ variables and the task, under some sufficiently generic assumptions, is to estimate both s and A using the measured data.

The ICA approach assumes that the components $S_1, \ldots, S_n$ are statistically independent and have nongaussian (unknown) distributions. ICA is closely related to, and one way of, performing blind source separation (BSS) [62, 71]. In BSS, as in the ICA based methods, the source signal is recovered from the observed signals without prior knowledge. It is also assumed that the sources components are statistically independent. The ICA based BSS method relies on the Central Limit Theorem of statistics and its implication in the algorithm development, in particular the nongaussianity and independence of the sources components implications.

The ICA approach relies on the basic principle of finding an estimator that maximises the nongaussianity and hence independence of the estimated variables and therefore provides an estimator that gives close approximation for the $S_1, \ldots, S_n$ components given the measured $X_1, \ldots, X_n$ variables. Given that the mixing matrix A is unknown, the estimated source components values are not exact. In particular, for each component there are two local maxima with negative and positive values and hence the ICA approach provides optimal estimates up to an ambiguous sign and amplitude. In the cases where the sources components are all Gaussian random variable, ICA fails to estimate the sources components.

Some of the early applications of the ICA in image processing and denoising can be found in [72–74]. Bell and Sejnowski compared the ICA based image filters with other decorrelating filters and found out that the ICA based filters have more sparsely distributed (nongaussian) outputs on natural scenes. Hyvarinen et al. [72] propose a method for image denoising based on ICA where the denoising parameters need to be learned from noise free images. Qingfu et al. [73] propose a simplified method based on Hyvarinen et al.'s denoising method in which only one heuristic parameter is used and no noise free images are required. From the above, it can be seen that ICA is a nongaussianity based approach that recovers the original signal/image from the observed signal/image without prior knowledge and with only very general assumption about the source data. The statistical independence of the sources signals is assumed and is fundamental in this approach. The ICA approach to signal separation and denoising would fail in the cases where the statistical distributions of the sources signals are all Gaussian. The ICA approach relies on the principle that the solution of the ICA problems is the recovered source signals/images with the maximum nongaussianity. Therefore, finding a combination of the observed signal/image that has the maximum nongaussianity solves the problem and restores the independent source signal/image. When the original signals are i.i.d (Identically Independent distributions) over space, blind deconvolution of images becomes a special case of ICA with only one observed image that consists of one unknown source image (original) mixed with itself at different spatial location [67]. Using ICA principles for signal/image restoration requires a preprocessing step in which the observed signal/image is centred and normalised before applying the nongaussianity measures.

Umeyama [68] presented a blind image deconvolution method based on ICA and its principles. The algorithm of this method starts by processing the input blurred image with Gabor filters and the resulting images from these filters and the source blurred image are then fed into the ICA algorithm. Gabor filters have characteristics that resembles certain characteristics of the cells in the visual cortex of some mammals. These filters decompose an input image into a number of filtered differential images. The implemented ICA algorithm in this method uses an online stochastic gradient method to estimate the weight vector and hence the original image. The application of the Gabor filters on the input blurred image results in a number of images that are derivative of the original image and are used as an input into the ICA. These images and the original blurred image are fed to the ICA algorithm. This method of image restoration does not constrain the shape of the convolution filter and it only assumes that the original image and its derivatives are independent components. The ICA algorithm used in this method starts by transforming the input images into vectors composed from the image pixels values. The image is scanned in a left-to-right and top-to-bottom manner. Then, the data is centred, to make it zero mean, and whitened. The reported implementation of the method assumes the PSF is uniform and is applied to artificial and real images without additive noise. It was reported that the proposed method works well for different shapes of PSF and the PSF estimation was fairly good.

In more recent development, Hujun et al. [67] state that the classical methods of image restoration are either domain specific, applicable for a particular type of blurring function or require prior knowledge; they also state that gradient based image deblurring methods can be easily trapped in local minima. Hujun et al. [67] present a method for blind image deconvolution and deblurring based on the Central Limit Theorem, the ICA principles, nongaussianity measures and genetic algorithms. The presented scheme is based on a single image and assumes no prior knowledge about the image, the blurring function or the noise. Experiments were conducted on spatial invariant point spread functions PSFs with blurring image. The task of the developed method is to find the deblurring filter that can be deconvoluted with the deblurring image to restore the true image.

The algorithm starts with genetic parameters initialization and a measurement of the blurred image nongaussianty before the image is transformed into the frequency domain. In the frequency domain, an optimal blurring kernel is searched for in combination with frequency domain image restoration through inverse/wiener filtering. The restored image is then converted to the spatial domain to calculate its nongaussianity. The algorithm evolves the solution using the nongaussianity measure as the fittest function for the child population generation. The algorithm iterates until it converges or stops after a set number of iterations.

Hujun et al.'s method was tested on artificially blurred images with atmospheric turbulence blur, uniform out of focus blur and linear motion blur. With the use of PSNR, the performance of this method is compared with the standard methods; Wiener filter, Regularized filter, Richardson-Lucy method and Blind Richardson-Lucy method. The comparison, in most cases, shows a better performance in terms of PSNR. Hujun et al.'s method depends on the nongaussianity measure in evolving the solution to convergence and hence its performance in natural images depends on the performance of the nongaussianity measure in the presence of outliers and on the level of noise present in the blurred image.

Kurtosis in the spatial domain suffers from deblurring noise and ringing artefacts [11]. Therefore the robustness of BID schemes, based on spatial domain kurtosis nongaussianity measure, becomes unpractical. To overcome these issues, Khan et al. [75] developed a new nongaussianity measure, spectral Kurtosis, and presented a BID scheme based on the statistical independence of the of true image and the blurring PSF. The scheme uses the developed spectral kurtosis measure in combination with a GA to optimise the spectral kurtosis in the frequency domain rather than in the spatial domain. This method assumes, as is the case in the ICA technique, that the sources are statistically independent. In this case, the sources are the true image and the filter function (PSF). Assuming the true image is non-Gaussian, the developed scheme tries to maximise the nongaussianity of the deblurred image and hence increase the independence of the blurred image sources. It is reported [75] that the developed spectral kurtosis quality measure maximises at close or at the true point spread function. Khan and Yin state that the spectral domain approach to image deblurring offers several advantages over the spatial domain. In particular, the spectral kurtosis image quality measure is independent of distortion features such as ringing and noise and does not require reference image. Khan [11] also reports that the scheme offers several advantages over the spatial kurtosis based scheme in terms of computational efficiency and robustness.

As the classical quality performance measures of image restoration are based on errors between the restored image and the reference image [76], these measures are not applicable when a reference image is not available and hence not applicable in deblurring schemes that incorporate performance maximization criteria. One solution for this is issue is proposed by Khan et al. [76]. Khan et al. proposed a scheme, see 2.15, of measuring the improvement of the deblurred image quality by reblurring the restored image using a candidate PSF and then using PSNR between the original blurred image and the reblurred image. Evaluation tests for the scheme were conducted with noiseless blurred images and Weiner Filter to compute the deblurred image from a candidate PSF generated on the basis of the PSNR values. It is reported that the Wiener filter required manual adjustment to tune the NSR parameter to reduce the noise level in the deblurred images. Khan et al. [76] report that in comparison to spatial kurtosis measure, the PSNR measure of the blurred and reblurred images is less error prone to noise and ringing artefacts arising from the Wiener filter.

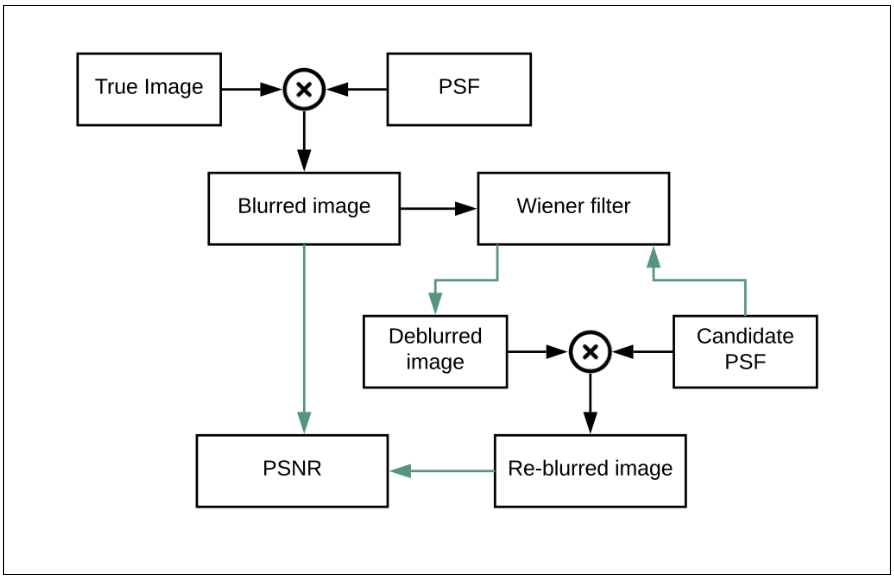


FIGURE 2.15: schematic diagram of the non-gaussianity based deblurring scheme.

Nongaussianity schemes to BID are based on the statistical assumption of the independence of the sources signals that combine to produce the measured signal/image and the statistical central limit theorem. This approach to blind source separation and image restoration has received considerable attention in the last two decades and several schemes for blurred images restoration are proposed in the literature. Some of these schemes use the ICA algorithms as one of the deblurring processing stages while others use the nongaussianity measures to evolve the solutions to convergence. The robustness or performance of these methods is affected by the used nongaussianity measure and the optimisation technique. Robust nongaussianity quality measures are developed and the application of these quality measures in BID schemes is reported.

2.4.10 Approximate Greatest Common Divisor (AGCD)

Linear algebra was first used to deblur images by Pillai and Liang [18, 19]. They used the greatest common divisor (GCD) method to extract the original image from two blurred versions of the true image without assuming a priori knowledge about the PSF. In this method, the real image is the GCD between the two blurred versions of the true image. In particular, they used a one-dimensional Sylvester resultant matrix algorithm to compute the GCD. The literature review has shown that there has been little further development of the solution of the BID problem using the GCD method since then. Statistical methods, namely Bayesian, have been used by many researchers in image restoration, as discussed previously. These methods suffer from no-blur solutions which are a consequence of the dependence of the methods on priori assumptions about the blurring function and the noise level [56].

In 2005, Raymond Heindl [77] introduced a slightly improved version of Pillai and Liang's method. The GCD method in its original form suffers from high sensitivity to noise and also using the Sylvester matrix is computationally expensive. Therefore, Li et al. in 2010 [1, 2] introduced a new method based on the Bezout matrix in order to develop an efficient GCD algorithm. Along with constraining the solution, they developed the BID algorithm based on the GCD algorithm for univariate polynomials and the Fast Fourier Transform (FFT). However, despite their method based on the Bezout matrix being less computationally expensive compared to the developed method based on the Sylvester matrix, the result obtained from the developed method based on the Sylvester matrix is more robust in the presence of noise than the result obtained from their method (see Chapter 6).

In 2012, Danelakis et al. [38] used the approximate greatest common divisor (AGCD) instead of GCD by introducing the fast upper triangularisation of the modified Sylvester matrix. However, the limitation of their method is the weakness of work on the large size of the two-dimensional PSF. Their method works on a separable and three-by-three PSF. They assume that the three-by-three PSF is symmetric with respect to its middle elements. Thus, the first and last rows of the PSF are computed as the AGCD of two selected rows of the blurred image. Then, the first and last columns of the PSF are computed as the AGCD of two selected columns of the blurred image, as well. After that, in order to calculate the middle element, they use the common property of PSF, which is the sum of all PSF elements equal to 1. Therefore, the middle element is computed by subtracting from 1 the sum of the AGCD's coefficients already found.

In describing all these methods and their limitations, the intention has been to answer one question: ''Is it possible to develop a new BID method that can recover the PSFs, separable or non-separable, that is not sensitive to additive noise?". Therefore, the current research has been conducted to develop a new BID method that can overcome almost all of the limitations detailed in the literature. The next chapter demonstrates the proposed BID method, one which overcomes the limitations in the literature, and examines extent to which the proposed BID method works.

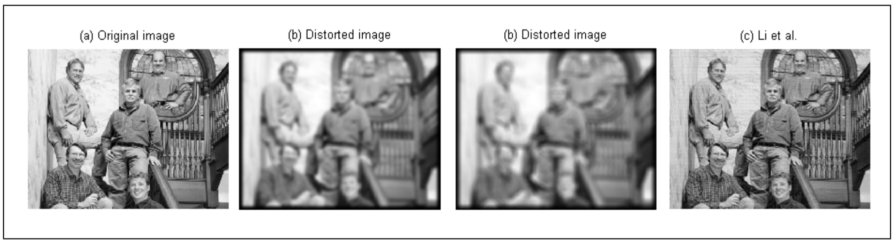


FIGURE 2.16: Image deblurring using the AGCD-based method developed by Li et al. [1, 2]. (a) a clear image; (b) the first blurred image caused by Gaussian blur; (c) the second blurred image caused by out-of-focus blur; (d) the restored image.

2.5 The Measurement of Image Performance

In order to compare the results of the developed method with the state-of-the-art methods, the measurement of image performance for BID must be considered. Often, the differences between two images are measured using signal-to-noise ratio (SNR), the peak signal to signal noise ratio ($PSNR$), mean squared error (MSE), and root mean squared error (RMSE).

The formulae of these measurement procedures are listed below. Let K be the exact image and W be the estimated version of K from the blurred version of K. The images are of size $M \times N$. The equations used for each of the above procedures are given in the following:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[K(i,j) - W(i,j)]^2, \tag{2.36}$$

$$RMSE = \sqrt{MSE}, \tag{2.37}$$

where the MSE and RMSE are the mean square errors between the true image signal and the noise signal, while the SNR is the power ratio between meaningful information (true image data) and unwanted information (noise)

$$SNR = \frac{10}{\log 10} \log \left\{ \frac{[\max_{i,j}\{K(i,j) - W(i,j)\}]^2}{\sum\limits_{i=1}^{M} \sum\limits_{j=1}^{N} [K(i,j) - W(i,j)]^2} \right\}. \tag{2.38}$$

PSNR performance measure is the ratio of the square of the maximum signal intensity. In addition to the above, the performance criterion of Relative Error is also used. Relative Error is defined as the absolute value of the error between the estimated image and the true exact image divided by the absolute value of the exact image.

$$\textsc{Relative error} = \frac{\|W - K\|_F}{\|K\|_F}, \tag{2.39}$$

where the subscript F denotes the Frobenius norm. Relative error is used when an indication of how good a measurement is in relation to its actual measured image. This error takes into account the size of the exact image when it is computed. In particular, a moderate error compare to small size of image would give a significant value; however, a moderate error value when compared to a large exact image value may not give a significant value. Hence, Relative Error is considered to be a suitable estimation performance measure for image restoration in this research study.

This research study uses some of the above performance criteria to compare the performance of the developed method and the state-of-the-art methods in recovering blurred images.

2.6 Summary

This chapter has reviewed the basic concepts of image deconvolution, including the problem formulation. This is followed by a brief introduction to PSF including a definition, description of some of the properties of the blurring function, and the most commonly occurring types of blur.

The serious distortion of the image comes from the PSF which is a function that describes the procedure of blurring an image. There are two type of the PSF. The first one is known as a spatially invariant blur in which case the influence of the blur is the same for every pixel in the blurred image. The second one is known as a spatially variant blur in which case the blur varies over the image. In general, The blurring in images arises from many sources during the capture process. For example, it can arise from limitations of the optical system, camera and object motion, defocused objects, lens error, and environmental effects such as atmospheric perturbations. The most common types of blur are Gaussian blur, motion blur and out-of-focus blur.

Furthermore, some image restoration filters are described, starting with basic image restoration filters followed by the most recent approaches. These image restoration filters and the BID methods can be listed as follows: Inverse Filtering, Wiener Filtering, Iterative Blind Deconvolution Method, Richardson-Lucy Algorithm, Regularisation-Based Deblurring Algorithm, Maximum Likelihood Restoration Method, Total Variation Blind Deconvolution, Maximum a Posteriori (MAP) based method, Nongaussianity Approach to Blind Image Deconvolution, and the Approximate Greatest Common Divisor (AGCD) to Blind Image Deconvolution. Functionalities, advantages, limitations, and examples of all the discussed image restoration filters and the BID method are provided at this chapter.

The measurements of image quality are detailed at the end of this chapter. The following chapter will illustrate the most important tools used in the proposed BID method.

Chapter 3

The Approximate Greatest Common Divisor method for image restoration

3.1 Introduction

The determination of the true image requires calculating the blurring function (PSF). Removing the blurring function from the blurred image is considered to be more significant than removing noise [42]. Using linear algebra for solving the blind image deconvolution (BID) is connected to the computation of the greatest common divisor (GCD) of two bivariate polynomials, as was discussed in Section 2.4.10. It is important to define the GCD of two exact polynomials and the approximate greatest common divisor (AGCD) of two inexact polynomials before proceeding further with the image deblurring algorithm. This chapter provides a brief introduction of the GCD of two univariate polynomials. Then, the computation of the AGCD that has been used in this research is illustrated in this chapter. As has been mentioned, the problem of BID is an example of an ill-posed linear problem [42]. Thus, the AGCD computations must be done with care. Therefore, the AGCD computations are performed using an approximate polynomial factorisation by the Sylvester resultant matrix of two inexact polynomials that was developed by Winkler et al. [17]. A brief introduction of the Sylvester resultant matrix and its subresultant matrices is considered in the next section.

3.2 Greatest Common Divisor (GCD)

The computation of the GCD of two exact univariate polynomials is a basic method in the algebraic computation field. It can be defined easily by the following: Let $f(x)$ and $g(x)$ refer to the two exact polynomials of degrees m and n, respectively, and their coefficients are known exactly:

$$f(x) = a_0x^m + a_1x^{m-1} + \cdots + a_m, \qquad a_0 \neq 0, \tag{3.1}$$

and

$$g(x) = b_0x^n + b_1x^{n-1} + \cdots + b_n, \qquad b_0 \neq 0, \tag{3.2}$$

where $a_0, a_1, \ldots, a_m$ and $b_0, b_1, \ldots, b_n$ are the coefficients of $f(x)$ and $g(x)$, respectively. The GCD of $f(x)$ and $g(x)$ is $d(x)$, is defined as a non-constant common divisor polynomial of the highest degree,

$$d(x) = \frac{f(x)}{u(x)} = \frac{g(x)}{v(x)}, \tag{3.3}$$

where $u(x)$ and $v(x)$ are quotient polynomials and are co-prime which means that their greatest common divisor is equal to 1 or nonzero constant. The computation of the degree and coefficients of GCD using the Sylvester resultant matrix is presented in the next section.

3.3 Sylvester Resultant Matrix

The Sylvester resultant matrix is a type of resultant matrix that is associated with two univariate polynomials. The Sylvester resultant matrix can be used to calculate the GCD and an AGCD of two polynomials. If the two polynomials are not co-prime, it is possible to calculate the degree and coefficients of their GCD using the Sylvester resultant matrix by two ways. The first method based on the Sylvester resultant matrix is that the degree of the GCD is equal to the rank loss of their Sylvester resultant matrix [78]. Then, the coefficients of the GCD can be obtained by reducing the Sylvester resultant matrix to an upper triangular form through the LU or QR decompositions [79]. The LU decomposition is a standard method of decomposing matrices. In this decomposition, the matrix is decomposed into a product of two triangular matrices where the first is lower triangular and the second is upper triangular. Lower triangular matrices have all the elements above the diagonal which are zeros whereas upper triangular matrices have all the elements below the diagonal which are zeros. While, QR decomposition is a method of decomposing a matrix into a product of two matrices where the right hand side of the product is an orthogonal matrix Q and the left hand side of the product is an upper triangular matrix R. Therefore, the coefficients of GCD are contained in the last non-zero row of the matrices U or R, respectively. Unfortunately, this computation of GCD using the Sylvester resultant matrix becomes more complicated when the two polynomials become relatively prime (co-prime) regarding the additive noise that perturbs the coefficients of these polynomials.

The second method based on the Sylvester resultant matrix is that the degree and coefficients of the GCD are computed using the Sylvester resultant matrix and its subresultant matrix (see Theorem 3.1)

Theoretically, the coefficients of these polynomials with additive noise can transform the singular matrix to a non-singular matrix. In that case, the Sylvester resultant matrix of these polynomials has full rank. Consequently, it is necessary to implement some pre-processing operations on the two inexact polynomials before using the Sylvester resultant matrix [17]. These operations are considered in the next section.

The Sylvester resultant matrix of two exact polynomials can be briefly introduced as follows: Assume that $f = f(x)$ and $g = g(x)$ refer to the two exact polynomials of degrees m and n, respectively, equation (3.1) and(3.2) can be rewritten as follows:

$$f(x) = \sum_{i=0}^{m} a_i x^{m-i}, \qquad a_0 \neq 0, \tag{3.4}$$

and

$$g(x) = \sum_{i=0}^{n} b_i x^{n-i}, \qquad b_0 \neq 0. \tag{3.5}$$

Then, the Sylvester resultant matrix $S(f,g) \in \mathbb{R}^{(m+n)\times(m+n)}$ of the two polynomials $f(x)$ and $g(x)$ is given by:

$$S(f,g) \quad = \quad \underbrace{\left[\begin{array}{cccc|cccc} a_0 & & & & b_0 & & & \\ a_1 & a_0 & & & b_1 & b_0 & & \\ \vdots & a_1 & \ddots & & \vdots & b_1 & \ddots & \\ a_{m-1} & \vdots & \ddots & a_0 & b_{n-1} & \vdots & \ddots & b_0 \\ a_m & a_{m-1} & \ddots & a_1 & b_n & b_{n-1} & \ddots & b_1 \\ & a_m & \ddots & \vdots & & b_n & \ddots & \vdots \\ & & \ddots & a_{m-1} & & & \ddots & b_{n-1} \\ & & & a_m & & & & b_n \end{array}\right]}_{n \text{ columns} \qquad\qquad m \text{ columns}}, \tag{3.6}$$

The first n columns are occupied by the coefficients a_i of $f(x)$, and the last m columns are occupied by the coefficients b_i of $g(x)$, and each of the two sub-matrices is Toeplitz.

The Sylvester resultant matrix and its subresultant matrices are used in order to compute the GCD of two exact polynomials. The following theorem shows the computation of the degree and coefficient of GCD of f and g using the Sylvester subresultant matrix.

Theorem 3.1. *[80, 81] The exact polynomials $f(x)$ and $g(x)$ have a common divisor of degree $k \geq 1$ if, and only if, the rank of its Sylvester subresultant matrix $S_k(f, g)$ is less than or equal to $m+n-2k+1$. The degree of the common divisor of $f(x)$ and $g(x)$ is equal to $k = 1, \ldots, t$, and there is no common divisor of degree $k \geq t+1$ if, and only if, the rank of $S_k(f, g)$ satisfies:*

$$\begin{aligned} rank\ S_k(f,g) &\leq m+n-2k+1, && k = 1, \ldots, t, \\ rank\ S_k(f,g) &= m+n-2k+2, && k = t+1, \ldots, \min(m,n), \end{aligned} \tag{3.7}$$

Proof. Since the polynomials $f(x)$ and $g(x)$ are not co-prime, which means that they have a non-constant common divisor polynomial $d_k(x)$ of the degree of k, as well as quotient polynomials $u_k(x)$ and $v_k(x)$, they will satisfy the following equations:

$$f(x) = d_k(x)u_k(x), \qquad \deg u_k < \deg f = m, \tag{3.8}$$

and

$$g(x) = d_k(x)v_k(x), \qquad \deg v_k < \deg g = n, \tag{3.9}$$

where at $d_k(x)$, $v_k(x)$, and $v_k(x)$ are polynomials given by

$$d_k(x) = \sum_{i=0}^{k} d_{k,i}x^{k-i}, \tag{3.10}$$

$$u_k(x) = \sum_{i=0}^{m-k} u_{k,i}x^{m-k-i}, \tag{3.11}$$

$$v_k(x) = \sum_{i=0}^{n-k} v_{k,i} x^{n-k-i}, \tag{3.12}$$

for all common divisors of degree $k = 1, \ldots, t$, where t is the greatest degree of the non-constant common divisor polynomials of $f(x)$ and $g(x)$. From (3.8) and (3.9),

$$d_k(x) = \frac{f(x)}{u_k(x)} = \frac{g(x)}{v_k(x)}. \tag{3.13}$$

Then it follows that

$$f(x)v_k(x) \quad = \quad g(x)u_k(x). \tag{3.14}$$

This equation can be written in matrix form as follows:

$$\begin{bmatrix} C_{n-k+1}(f) & D_{m-k+1}(g) \end{bmatrix} \begin{bmatrix} \mathbf{v}_k \\ -\mathbf{u}_k \end{bmatrix} = S_k \begin{bmatrix} \mathbf{v}_k \\ -\mathbf{u}_k \end{bmatrix} = 0, \qquad k = 1, \ldots, t, \tag{3.15}$$

where $C_{n-k+1}(f) \in \mathbb{R}^{(m+n-k+1)\times(n-k+1)}$ and $D_{m-k+1}(g) \in \mathbb{R}^{(m+n-k+1)\times(m-k+1)}$ are Toeplitz matrices of the coefficients of $f(x)$ and $g(x)$, and $\mathbf{v}_k$ and $\mathbf{u}_k$ are vectors of the coefficients of $v_k(x)$ and $u_k(x)$, respectively.

$$C_{n-k+1}(f) = \begin{bmatrix} a_0 & & \\ a_1 & \ddots & \\ \vdots & \ddots & a_0 \\ a_{m-1} & \ddots & a_1 \\ a_m & \ddots & \vdots \\ & \ddots & a_{m-1} \\ & & a_m \end{bmatrix}, \quad D_{m-k+1}(g) = \begin{bmatrix} b_0 & & \\ b_1 & \ddots & \\ \vdots & \ddots & b_0 \\ b_{n-1} & \ddots & b_1 \\ b_n & \ddots & \vdots \\ & \ddots & b_{n-1} \\ & & b_n \end{bmatrix}, \tag{3.16}$$

and

$$\mathbf{u}_k = \begin{bmatrix} u_{k,0} & u_{k,1} & \ldots & u_{k,m-k-1} & u_{k,m-k} \end{bmatrix}^T, \tag{3.17}$$

where $\mathbf{u}_k \neq 0$ for all $k = 1, \ldots, t$, and $\mathbf{u}_k \equiv 0$ for $k = t+1, \ldots, \min(m,n)$, and

$$\mathbf{v}_k = \begin{bmatrix} v_{k,0} & v_{k,1} & \ldots & v_{k,n-k-1} & v_{k,n-k} \end{bmatrix}^T, \tag{3.18}$$

where, as well, $\mathbf{v}_k \neq 0$ for all $k = 1, \ldots, t$ and $\mathbf{v}_k \equiv 0$ for $k = t+1, \ldots, \min(m, n)$.

The Toeplitz matrices $C_{n-k+1}(f)$ and $D_{m-k+1}(g)$ are expressed in one Sylvester resultant matrix $S = S(f, g)$, in which $S_k = S_k(f, g) \in \mathbb{R}^{(m+n-k+1)\times(m+n-2k+2)}$ is the kth Sylvester subresultant matrix, which is formed by deleting some rows and columns from $S_k = S_k(f, g)$. The kth Sylvester subresultant matrix $S_k(f, g)$ is obtained by deleting the last $(k-1)$ rows of its matrix and the last $(k-1)$ columns of $C_{n-k+1}(f)$ and $D_{m-k+1}(g)$. Theoretically, it is clear that the Sylvester subresultant matrix for the index $k = 1$ yields the Sylvester resultant matrix $S_1(f, g) = S(f, g)$.

The $S_k(f, g)$ is singular for $k \leq t$ and non-singular if, and only if, f and g are co-prime. In other words, there are three cases for the rank of the coefficient matrix in (3.15):

- There is more than one solution of (3.15) for $k = 1, \ldots, t-1$.
- There is one solution of (3.15) for $k = t$.
- There is no solution of (3.15) (apart from the zero solution) for $k \geq t+1$.

The next section discusses the three operations that must be processed before the Sylvester matrix of the two inexact polynomials is used to compute an AGCD.

3.4 Pre-processing Operations

This section discusses the pre-processing operations that must be applied to the given polynomials before their GCD is computed [17]. Section 3.3 has shown that the Sylvester resultant matrix and its subresultant matrices $S_k(f, g)$ can be used to determine the degree of the GCD of two exact $f(x)$ and $g(x)$. In contrast, the situation is different when two inexact polynomials $\hat{f}(x)$ and $\hat{g}(x)$ are considered because the matrix $S_k(\hat{f}, \hat{g})$

is reduced to a matrix of full rank because $\hat{f}(x)$ and $\hat{g}(x)$ are co-prime because of the noise that is present in $\hat{f}(x)$ and $\hat{g}(x)$.

In order to perform successive approximate GCD computations of two inexact polynomials $\hat{f}(x)$ and $\hat{g}(x)$, three pre-processing operations are applied for the improvement of computational results on rank estimation. These three pre-processing operations have been discussed in [80–82]. The first operation normalises each polynomial by the geometric mean of its coefficients. The second and the third operations minimise the ratio of the maximum coefficient in magnitude to the minimum coefficient in magnitude using two parameters α and θ that will be determined. α is a parameter that originates from the partitioned structure of the Sylvester matrix, and a parameter θ is used in order to scale the independent variable x. The parameter θ has been introduced in order to transform the normalised forms of the inexact polynomials $\hat{f}(x)$ and $\hat{g}(x)$ into another set of polynomials, whose coefficient variations are smaller.

The next subsections illustrate extensively the three pre-processing operations that require successive approximate GCD computations.

3.4.1 Normalisation

It was mentioned in Section 3.3 that the Sylvester matrix $S_k(\hat{f}, \hat{g})$ has a partitioned structure because the first $n-k+1$ columns are occupied by the coefficients of $\hat{f}(x)$, whereas the last $m-k+1$ columns are occupied by the coefficients of $\hat{g}(x)$. This partitioned structure may lead to the Sylvester matrix $S_k(\hat{f}, \hat{g})$ being unbalanced, especially if coefficients $\hat{f}(x)$ are significantly larger or smaller than the coefficients of $\hat{g}(x)$. More precisely, if coefficients $\hat{f}(x)$ are much larger than the coefficients of $\hat{g}(x)$, then $|\hat{a}_i| \gg |\hat{b}_j|, i = 0, \dots, m, j = 0, \dots, n$, and the rank of $S(\hat{f}, \hat{g})$ is approximately equal to n even if $\hat{f}(x)$ and $\hat{g}(x)$ are co-prime. On the other hand, if the coefficients $\hat{f}(x)$ are much smaller than the coefficients of $\hat{g}(x)$, then the rank of $S(\hat{f}, \hat{g})$ is approximately equal to m. Therefore, it is necessary to take precautions to address this problem by normalising the polynomials.

Consequently, the geometric mean of the coefficients of both polynomials $\hat{f}(x)$ and $\hat{g}(x)$ was used to provide better balance. The literature shows that the two-norm mean of the coefficient polynomials has been used frequently for normalisation for better conditioned [81, 83, 84]. However, it is shown that it is advantageous to normalise by using the geometric mean instead of other norms, for example, $1-, 2-$, and ∞ norm, because it provides a better and more balanced average when the coefficients of a polynomial vary widely in magnitude [80, 81, 84].

Thus, $\bar{f}(x)$ and $\bar{g}(x)$ are scaled from $\hat{f}(x)$ and $\hat{g}(x)$ by the geometric means of their coefficients, and are given by

$$\bar{f}(x) = \sum_{i=0}^{m} \bar{a}_i x^{m-i}, \qquad \bar{a}_i = \frac{\hat{a}_i}{\left(\prod_{i=0}^{m} |\hat{a}_i|\right)^{\frac{1}{m+1}}} \tag{3.19}$$

and

$$\bar{g}(x) = \sum_{i=0}^{n} \bar{b}_i x^{n-i}, \qquad \bar{b}_i = \frac{\hat{b}_i}{\left(\prod_{i=0}^{n} |\hat{b}_i|\right)^{\frac{1}{n+1}}} \tag{3.20}$$

where $\bar{a}_i$ and $\bar{b}_i$ are the normalised coefficients, and $\hat{a}_i$ and $\hat{b}_i$ are the non-normalised coefficients of $\hat{f}(x)$ and $\hat{g}(x)$, respectively; in the geometric means, especially, it is assumed that the coefficients are non-zero or the normalisation will be unstable. If, however, one or more of these coefficients are zero, then the geometric mean will be computed with respect to the non-zero coefficients only, rather than all of them.

3.4.2 Relative scaling of polynomials

The second pre-processing operation is introduced as follows:

$$GCD(\hat{f}, \hat{g}) \sim GCD(\hat{f}, \alpha\hat{g}), \tag{3.21}$$

where α is an arbitrary non-zero scalar multiplier and $\sim$ denotes equivalence to an arbitrary non-zero scalar multiplier. Thus, the polynomial $\hat{g}(x)$ can be generalised to $\alpha\hat{g}(x)$. Since inexact polynomials are considered, this equivalence fails numerically because different scalar values of α may lead to different AGCDs. In order to obtain a good computational result for AGCD, the variable α has to be used as a parameter that will be computed according to a specific criterion such that good results are obtained [80, 81]. According to the normalised forms of $\hat{f}(x)$ and $\hat{g}(x)$ in (3.19) and (3.20), respectively, α can be construed as the relative weight of $\hat{g}(x)$ to the unit weight of $\hat{f}(x)$.

The inclusion of the parameter α was first introduced in [85], where it was shown that the rank of $S(\hat{f}, \alpha\hat{g})$ is difficult to obtain by using a random value of α and not all the values of α are associated with a well-defined rank of $S(\hat{f}, \alpha\hat{g})$. Consequently, it is important to carefully choose the procedure of computing an optimal value of α in order to obtain a good approximation for an AGCD [80]; this will be illustrated in Section 3.4.4.

3.4.3 Scaling the independent variable

The third pre-processing operation will introduce the parameter θ that scales the independent variable x. This is achieved by the substitution

$$x = \theta y, \tag{3.22}$$

where θ is a parameter that will be determined to minimise the ratio of the maximum coefficient in magnitude to the minimum coefficient in magnitude, and y is the new independent variable. This operation is important because polynomials computations will be unreliable if the coefficients of polynomials have a significant variation in magnitude [86, 87]. The polynomials $\bar{f}(x)$ and $\bar{g}(x)$ in (3.19) and (3.20), respectively, are transformed to the polynomials $\tilde{f}(y)$ and $\tilde{g}(y)$ that are represented as

$$\tilde{f}(y) = \sum_{i=0}^{m} (\bar{a}_i \theta^{m-i}) y^{m-i}, \tag{3.23}$$

and

$$\tilde{g}(y) = \sum_{i=0}^{n} (\bar{b}_i \theta^{n-i}) y^{n-i}. \tag{3.24}$$

Once the AGCD of the two polynomials, $\tilde{f}(y)$ and $\tilde{g}(y)$, is computed, the polynomial of the GCD $\tilde{d}(y)$ is re-expressed in terms of the independent variable (x) by using the relationship $y = x/\theta$.

The method used to calculate the values of the parameters α and θ is studied in [84] and it has been extended into two polynomials in [80, 81]. The rest of this section discusses the procedure of choosing the optimal values α_0 and θ_0 of α and θ.

3.4.4 Calculating optimal values of the scaling parameters

The optimal values α_0 and θ_0 of α and θ, respectively, have been computed using the minimisation problem that has been studied in [17]. This computation is important to compute a good approximation of the GCD of $S_k(\tilde{f}_\theta(x), \alpha\tilde{g}_\theta(x))$ of the two polynomials $\tilde{f}(x)$ and $\tilde{g}(x)$. Therefore, the optimal value α_0 and θ_0 of α and θ can be determined simultaneously, such that the ratio of the maximum coefficient in magnitude to the minimum coefficient in magnitude of $S(\tilde{f}, \alpha\tilde{g})$, whose entries are $\{\bar{a}_i\theta^{m-i}\}_{i=0}^{m}$ and $\{\alpha\bar{b}_j\theta^{n-j}\}_{j=0}^{n}$, is minimised,

$$\alpha_0, \theta_0 = \arg\min_{\alpha,\theta} \left\{ \frac{\max\left\{\max_{i=0,\ldots,m} \left|\bar{a}_i\theta^{m-i}\right|, \max_{j=0,\ldots,n} \left|\alpha\bar{b}_j\theta^{n-j}\right|\right\}}{\min\left\{\min_{i=0,\ldots,m} \left|\bar{a}_i\theta^{m-i}\right|, \min_{j=0,\ldots,n} \left|\alpha\bar{b}_j\theta^{n-j}\right|\right\}} \right\}. \tag{3.25}$$

and this minimisation is converted into a linear programming problem (LP) [17, 88]. The solution of the LP problem yields the polynomials and all the GCD computations for the calculation of the PSF are reformed on these polynomials. It can represent the two polynomials as follows:

$$\tilde{f}_\theta(y) = \sum_{i=0}^{m} (\bar{a}_i \theta_0^{m-i}) y^{m-i}, \tag{3.26}$$

and

$$\tilde{g}_\theta(y) = \sum_{i=0}^{n} (\bar{b}_i \theta_0^{n-i}) y^{n-i}. \tag{3.27}$$

Then it will modify the Sylvester resultant matrix and its subresultant matrix of the given polynomials $\hat{f}(x)$ and $\hat{g}(x)$ after they are processed to $S(\tilde{f}_\theta(y), \alpha\tilde{g}_\theta(y))$; this can be written as follows:

$$S(\tilde{f}_\theta, \alpha\tilde{g}_\theta) \;=\; \begin{array}{c} \left[\begin{array}{cccc|cccc}
\bar{a}_0\theta_0^m & & & & \alpha_0\bar{b}_0\theta_0^n & & & \\
\bar{a}_1\theta_0^{m-1} & \bar{a}_0\theta_0^m & & & \alpha_0\bar{b}_1\theta_0^{n-1} & \alpha_0\bar{b}_0\theta_0^n & & \\
\vdots & \bar{a}_1\theta_0^{m-1} & \ddots & & \vdots & \alpha_0\bar{b}_1\theta_0^{n-1} & \ddots & \\
\bar{a}_{m-1}\theta_0 & \vdots & \ddots & \bar{a}_0\theta_0^m & \alpha_0\bar{b}_{n-1}\theta_0 & \vdots & \ddots & \alpha_0\bar{b}_0\theta_0^n \\
\bar{a}_m & \bar{a}_{m-1}\theta_0 & \ddots & \bar{a}_1\theta_0^{m-1} & \alpha_0\bar{b}_n & \alpha_0\bar{b}_{n-1}\theta_0 & \ddots & \alpha_0\bar{b}_1\theta_0^{n-1} \\
 & \bar{a}_m & \ddots & \vdots & & \alpha_0\bar{b}_n & \ddots & \vdots \\
 & & \ddots & \bar{a}_{m-1}\theta_0 & & & \ddots & \alpha_0\bar{b}_{n-1}\theta_0 \\
 & & & \bar{a}_m & & & & \alpha_0\bar{b}_n
\end{array}\right] \\
\underbrace{\hspace{6cm}}_{n \text{ columns}} \quad \underbrace{\hspace{6cm}}_{m \text{ columns}}
\end{array} . \tag{3.28}$$

This Sylvester matrix and its subresultant matrices are used for the computation of an approximate GCD of two inexact polynomials $S_k(\tilde{f}_{\theta_0}, \alpha_0\tilde{g}_{\theta_0})$. This computation is considered in the next section.

3.5 Approximate Greatest Common Divisor (AGCD)

The computation of the GCD of two polynomials arises in several applications, for example, computer-aided geometric design [89, 90], computer vision [91, 92], and computer graphics [93]. Euclid's algorithm is the classical method for the calculation of the GCD of two polynomials, but it cannot be used to calculate an AGCD of two polynomials. Practically, the given polynomials $\hat{f}$ and $\hat{g}$ are usually defined by inexact polynomials and the coefficients of polynomials are corrupted by noise since the computation of their GCD is performed in a floating point environment. Consequently, it is impossible to calculate the GCD of two inexact polynomials but it is possible to compute only an AGCD. More precisely, theoretically, the two inexact polynomials may have a non-constant GCD but the additive noise makes these polynomials co-prime. Therefore, it is necessary to perturb these inexact polynomials in order to induce a non-constant GCD. Then, the AGCD of the given inexact polynomials can be computed. There are several different AGCDs of two inexact polynomials corresponding to the criterion that is used for their calculation, for example, the nearest polynomial of given maximum degree or the polynomial of maximum degree within a given error tolerance [17].

Several methods have been developed in order to compute an AGCD of two polynomials: for instance, modifications to Euclid's algorithm [94], singular value decomposition [95], and the QR decomposition [96]. The Sylvester matrix and approximate polynomial factorisation (APF) have been considered in [97]. In this research, the developed method that has been applied to calculate the AGCD of two inexact polynomials $\hat{f}(x)$ and $\hat{g}(x)$ is the approximate polynomial factorisation (APF).

$$\hat{f}(x) \approx u_k(x)d_k(x) \qquad \hat{g}(x) \approx v_k(x)d_k(x), \tag{3.29}$$

where $d_k(x)$ is a non-constant common divisor polynomial of the degree of k, and $u_k(x)$ and $v_k(x)$ are quotient polynomials.

The APF requires the degree of an AGCD to be known. Therefore, the computation of an AGCD is divided into two stages: the first stage involves calculating the degree of an AGCD of two inexact polynomials using the Sylvester resultant matrix $S(\hat{f}, \hat{g})$. The second stage involves calculating the coefficients of an AGCD of their polynomials using approximate polynomial factorisation (APF). This section will present these two stages theoretically. It is noted that the two given inexact polynomials $\hat{f}(x)$ and $\hat{g}(x)$ will be transformed into $\tilde{f}_\theta(y)$ and $\tilde{g}_\theta(y)$ after applying the three pre-processed operations discussed in Section 3.4.

3.5.1 The computation of the degree of an AGCD

The computation of the degree of the AGCD constitutes an important step in the computation of the AGCD. Small errors in the computed degree will have a great effect on the computed AGCD polynomial.

The degree of the AGCD requires the computation of the rank of the Sylvester matrix and its sub-resultant matrices, as shown by Theorem 3.1. Existing methods for the calculation of the degree of the AGCD vary in their estimation accuracy from good to poor. The SVD method used in conjunction with the AGCD$(\tilde{f}_\theta, \alpha\tilde{g}_\theta)$ does not give an accurate estimation of the degree [17, 97] that is required by the AGCD. Thus, the principal angle and residual methods are used in [17, 88, 97] and they give an accurate estimation of the degree. However, this accuracy comes at a high cost in terms of computational time, which would limit its practical usage. Because these methods depend on the computation of the SVD of the Sylvester matrix of the two polynomials and all its sub-resultant matrices $k = 1, \ldots, \min(m, n)$, this means that the computation of the SVD of the Sylvester matrix of $(\tilde{f}_\theta, \alpha\tilde{g}_\theta)$ must be computed $\min(m, n)$ times. In contrast, the update formula for the QR decomposition is more efficient than its equivalent for the SVD. Thus the degree of the AGCD of two polynomials should be computed using QR decomposition, not the SVD. The SVD is a standard tool to calculate the rank for a matrix but the update formula for the QR decomposition is efficient in which the SVD and QR are cubic in complexity but the update formula for the QR in MATLAB is quadratic complexity. Thus, QR decomposition can be used to compute the degree of the AGCD. The QR decomposition is applied only to the Sylvester matrix.

Then, the QR of the sub-resultant matrices is achieved through the use of a simple update algorithm which involves a simple delete operation on the previous QR decomposition. In this research, the QR decomposition method is used for both its accuracy and computational efficiency.

To ensure the completeness of this research, the QR procedure developed by Bourne et al. [98] is described in detail in the following paragraphs.

Let $Q_k R_k$ be the QR decomposition of the Sylvester matrix and its sub-resultant matrices of $\tilde{f}_\theta(y)$ and $\tilde{g}_\theta(y)$ from $k = 1, \ldots, \min(m, n)$. For each R_k, define the diagonal elements by $r_{k,i,i}$ where $i = 1, 2, \ldots, m+n-2k+2$. The singularity of $S_k(\tilde{f}_\theta(y), \tilde{g}_\theta(y))$ is reflected by the value μ_k, because if:

$$\mu_k = \frac{\max_i |r_{k,i,i}|}{\min_i |r_{k,i,i}|}, \qquad k = 1, \ldots, \min(m, n), \tag{3.30}$$

then when the value of μ_k is close to 1, the sub-resultant matrix is of full rank, whereas when the value is much higher than one then the sub-resultant matrix is nearly singular. It follows that the degree t of an AGCD can be calculated using the values of μ_k because it follows from Theorem 3.1 and (3.30) that the μ_k values satisfy

$$\mu_k = \left\{ \begin{array}{cc} \infty, & k = 1, \ldots, t \\ \gamma_k < \infty, & k = t+1, \ldots, \min(m, n) \end{array} \right\}. \tag{3.31}$$

The μ_k values for singular R_k would be infinite while the μ_k for full-rank R_k will have finite values, because at least one diagonal entry of R_k is zero for $k = 1, \ldots, t$. Consequently, the degree of the AGCD can be found from the computed μ_k using the following criterion:

$$t = \arg \min_k (\mu_k - \mu_{k+1}), \qquad k = 1, \ldots, \min(m, n). \tag{3.32}$$

The computed AGCD degree is then used to compute the coefficient values of the AGCD polynomial, as described in the next section. It is important to note that these tests are heuristic, but many computational experiments have shown that they yield very good results for the degree of an AGCD of two inexact polynomials.

3.5.2 The computation of the coefficients of an AGCD using the Approximate Polynomial Factorization

The computation of the coefficients of an AGCD is considered in this section. Approximate Polynomial Factorization (APF) is used for computing the coefficients of an AGCD [17]. They used the structured non-linear total least norm (SNTLN) method for computation of a structured low-rank approximation of the Sylvester resultant matrix $S_d(\tilde{f}_\theta(y), \alpha\tilde{g}_\theta(y))$ of the two given inexact polynomials $\hat{f}(x)$ and $\hat{g}(x)$, where $\tilde{f}_\theta(y)$ and $\tilde{g}_\theta(y)$ are the two inexact polynomials $\hat{f}(x)$ and $\hat{g}(x)$ after the pre-processing operations that are defined in (3.26) and (3.27) respectively. The method of the structured nonlinear total least norm (SNTLN) is used to compute the coefficients of AGCD$(\tilde{f}_\theta(y), \alpha\tilde{g}_\theta(y))$ by applying it to the approximate polynomial factorisation of $\tilde{f}_\theta(y)$ and $\tilde{g}_\theta(y)$

$$\tilde{f}(y) \approx \tilde{u}_t(y)\tilde{d}_t(y) \quad \text{and} \quad \tilde{g}(y) \approx \tilde{v}_t(y)\tilde{d}_t(y) \tag{3.33}$$

where $\tilde{d}_t$ is an AGCD of degree t of the inexact polynomials $\tilde{f}_\theta(y)$ and $\tilde{g}_\theta(y)$, and $\tilde{u}_t(y)$ and $\tilde{v}_t(y)$ are the quotient polynomials. It is important to note that the degree of an AGCD (t) and the optimal value of the scaling θ_0 and α_0 are known using the methods described in Sections 3.5.1 and 3.4.4, respectively. It follows that

$$\tilde{d}_t = \tilde{d}_\theta(y) = \sum_{i=0}^{t} \left(\frac{\tilde{d}_{\theta,i}}{\theta_0^{t-i}} \right) \theta^{t-i} y^{t-i} = \sum_{i=0}^{t} \left(\tilde{r}_i \theta^{t-i} \right) y^{t-i}, \tag{3.34}$$

and

$$\tilde{u}_t = \tilde{u}_\theta(y) \quad = \quad \sum_{i=0}^{m-t} \left(\frac{\tilde{u}_i}{\theta_0^{m-t-i}} \right) \theta^{m-t-i} y^{m-t-i} = \sum_{i=0}^{m-t} \left(\tilde{c}_i \theta^{m-t-i} \right) y^{m-t-i} \tag{3.35}$$

$$\tilde{v}_t = \tilde{v}_\theta(y) \quad = \quad \sum_{i=0}^{n-t} \left(\frac{\tilde{v}_i}{\theta_0^{n-t-i}} \right) \theta^{n-t-i} y^{n-t-i} = \sum_{i=0}^{n-t} \left(\tilde{e}_i \theta^{n-t-i} \right) y^{n-t-i}, \tag{3.36}$$

where the quotient polynomials are co-prime, since

$$\tilde{r}_i = \frac{\tilde{d}_{\theta,i}}{\theta_0^{t-i}}, \quad \tilde{c}_i = \frac{\tilde{u}_{\theta,i}}{\theta_0^{m-t-i}}, \quad \text{and} \quad \tilde{e}_i = \frac{\tilde{v}_{\theta,i}}{\theta_0^{n-t-i}}. \tag{3.37}$$

It follows from (3.26), (3.27), (3.34), (3.35), and (3.36) that the full form of (3.33) is

$$\sum_{i=0}^{m} \left(\bar{a}_i \theta^{m-i} \right) y^{m-i} \approx \left(\sum_{i=0}^{m-t} \left(\tilde{c}_i \theta^{m-t-i} \right) y^{m-t-i} \right) \left(\sum_{i=0}^{t} \left(\tilde{r}_{t,i} \theta^{t-i} \right) y^{t-i} \right), \tag{3.38}$$

and

$$\alpha_0 \sum_{i=0}^{n} \left(\bar{b}_i \theta^{n-i}\right) y^{n-i} \approx \left(\sum_{i=0}^{n-t} \left(\tilde{e}_i \theta^{n-t-i}\right) y^{n-t-i} \right) \left(\sum_{i=0}^{t} \left(\tilde{r}_{t,i} \theta^{t-i}\right) y^{t-i} \right), \tag{3.39}$$

which can be written in matrix form

$$\begin{bmatrix} C_{t,1}(\tilde{c}_t, \theta) \\ C_{t,2}(\tilde{e}_t, \theta) \end{bmatrix} \tilde{\mathbf{r}}_t(\theta) \approx \begin{bmatrix} \tilde{\mathbf{f}}(\theta) \\ \alpha_0 \tilde{\mathbf{g}}(\theta) \end{bmatrix}, \tag{3.40}$$

where $C_{t,1}(\tilde{c}_t, \theta)$ and $C_{t,2}(\tilde{e}_t, \theta)$ are Toeplitz matrices.

$$C_{t,1}(\tilde{c}_t, \theta) = \begin{bmatrix} \tilde{c}_{t,0}\theta^{m-t} & & & \\ \tilde{c}_{t,1}\theta^{m-t-1} & \tilde{c}_{t,0}\theta^{m-t} & & \\ \tilde{c}_{t,2}\theta^{m-t-2} & \tilde{c}_{t,1}\theta^{m-t-1} & \ddots & \\ \vdots & \tilde{c}_{t,2}\theta^{m-t-2} & \ddots & \tilde{c}_{t,0}\theta^{m-t} \\ \vdots & \vdots & \ddots & \tilde{c}_{t,1}\theta^{m-t-1} \\ \tilde{c}_{t,m-t-1}\theta & \vdots & \ddots & \tilde{c}_{t,2}\theta^{m-t-2} \\ \tilde{c}_{t,m-t} & \tilde{c}_{t,m-t-1}\theta & \ddots & \vdots \\ & \tilde{c}_{t,m-t} & \ddots & \vdots \\ & & \ddots & \tilde{c}_{t,m-t-1}\theta \\ & & & \tilde{c}_{t,m-t} \end{bmatrix} \in \mathbb{R}^{(m+1)\times(t+1)}, \tag{3.41}$$

and

$$C_{t,2}(\tilde{e}_t,\theta) = \begin{bmatrix} \tilde{e}_{t,0}\theta^{n-t} & & & \\ \tilde{e}_{t,1}\theta^{n-t-1} & \tilde{e}_{t,0}\theta^{n-t} & & \\ \tilde{e}_{t,2}\theta^{n-t-2} & \tilde{e}_{t,1}\theta^{n-t-1} & \ddots & \\ \vdots & \tilde{e}_{t,2}\theta^{n-t-2} & \ddots & \tilde{e}_{t,0}\theta^{n-t} \\ \vdots & \vdots & \ddots & \tilde{e}_{t,1}\theta^{n-t-1} \\ \tilde{e}_{t,n-t-1}\theta & \vdots & \ddots & \tilde{e}_{t,2}\theta^{n-t-2} \\ \tilde{e}_{t,n-t} & \tilde{e}_{t,n-t-1}\theta & \ddots & \vdots \\ & \tilde{e}_{t,n-t} & \ddots & \vdots \\ & & \ddots & \tilde{e}_{t,n-t-1}\theta \\ & & & \tilde{e}_{t,n-t} \end{bmatrix} \in \mathbb{R}^{(n+1)\times(t+1)}, \qquad (3.42)$$

and

$$\begin{aligned} \tilde{\mathbf{f}}(\theta) &= \begin{bmatrix} \bar{a}_0\theta^m & \bar{a}_1\theta^{m-1} & \cdots & \bar{a}_{m-1}\theta & \bar{a}_m \end{bmatrix}^T \in \mathbb{R}^{m+1}, \\ \tilde{\mathbf{g}}(\theta) &= \begin{bmatrix} \bar{b}_0\theta^n & \bar{b}_1\theta^{n-1} & \cdots & \bar{b}_{n-1}\theta & \bar{b}_n \end{bmatrix}^T \in \mathbb{R}^{n+1}, \\ \tilde{\mathbf{r}}_t(\theta) &= \begin{bmatrix} r_{t,0}\theta^t & r_{t,1}\theta^{t-1} & \cdots & r_{t,t-1}\theta & r_{t,t} \end{bmatrix}^T \in \mathbb{R}^{t+1}. \end{aligned} \qquad (3.43)$$

The application of the method of SNTLN requires adding a structured matrix to the coefficient matrix on the left-hand side and a structured vector to the right-hand side of the approximate (3.40), in order to perturb the approximate equation to the equation that has an exact solution. Therefore, the approximate equation is replaced by

$$\begin{bmatrix} C_{t,1}(\tilde{c},\theta) + E_{t,1}(z_t,\theta) \\ C_{t,2}(\tilde{e},\theta) + E_{t,2}(z_t,\theta) \end{bmatrix} \tilde{\mathbf{r}}_t(\theta) = \begin{bmatrix} \tilde{\mathbf{f}}(\theta) + \mathbf{s}_t(p_t,\theta) \\ (\alpha_0+\beta_0)\left(\tilde{\mathbf{g}}(\theta) + \mathbf{w}_t(q_t,\theta)\right) \end{bmatrix}, \qquad (3.44)$$

where

$$E_{t,1}(z_t,\theta) \;=\; \begin{bmatrix} z_{t,0}\theta^{m-t} & & \\ z_{t,1}\theta^{m-t-1} & \ddots & \\ z_{t,2}\theta^{m-t-2} & \ddots & z_{t,0}\theta^{m-t} \\ \vdots & \ddots & z_{t,1}\theta^{m-t-1} \\ \vdots & \ddots & z_{t,2}\theta^{m-t-2} \\ z_{t,m-t-1}\theta & \ddots & \vdots \\ z_{t,m-t} & \ddots & \vdots \\ & \ddots & z_{t,m-t-1}\theta \\ & & z_{t,m-t} \end{bmatrix} \in \mathbb{R}^{(m+1)\times(t+1)}, \tag{3.45}$$

and

$$E_{t,2}(z_t,\theta) \;=\; \begin{bmatrix} z_{t,m-t+1}\theta^{n-t} & & \\ z_{t,m-t+2}\theta^{n-t-1} & \ddots & \\ z_{t,m-t+3}\theta^{n-t-2} & \ddots & z_{t,m-t+1}\theta^{n-t} \\ \vdots & \ddots & z_{t,m-t+2}\theta^{n-t-1} \\ \vdots & \ddots & z_{t,m-t+3}\theta^{n-t-2} \\ z_{t,m+n-2t}\theta & \ddots & \vdots \\ z_{t,m+n-2t+1} & \ddots & \vdots \\ & \ddots & z_{t,m+n-2t}\theta \\ & & z_{t,m+n-2t+1} \end{bmatrix} \in \mathbb{R}^{(n+1)\times(t+1)}. \tag{3.46}$$

$E_{t,1}$ and $E_{t,2}$ are Toeplitz matrices of the perturbations $z_{t,i}$

$$z_t = \begin{bmatrix} z_{t,0} & \cdots & z_{t,m-t} & z_{t,m-t+1} & \cdots & z_{t,m+n-2t+1} \end{bmatrix}^T \in \mathbb{R}^{m+n-2t+2}, \tag{3.47}$$

that are added to the coefficients $\tilde{c}_{t,i}$ and $\tilde{e}_{t,i}$, are given by the vectors $\mathbf{s}_t = \mathbf{s}_t(p_t, \theta) \in \mathbb{R}^{m+1}$ and $\mathbf{w}_t = \mathbf{w}_t(q_t, \theta) \in \mathbb{R}^{n+1}$,

$$\begin{aligned} \mathbf{s}_t &= \left[\begin{array}{cccccc} p_{t,0}\theta^m & p_{t,1}\theta^{m-1} & \cdots & p_{t,m-1}\theta & p_{t,m} \end{array}\right]^T \in \mathbb{R}^{m+1}, & (3.48) \\ \mathbf{w}_t &= \left[\begin{array}{cccccc} q_{t,0}\theta^n & q_{t,1}\theta^{n-1} & \cdots & q_{t,n-1}\theta & q_{t,n} \end{array}\right]^T \in \mathbb{R}^{n+1}, & (3.49) \end{aligned}$$

where

$$\begin{aligned} p_t &= \left[\begin{array}{ccccc} p_{t,0} & p_{t,1} & \cdots & p_{t,m-1} & p_{t,m} \end{array}\right]^T \in \mathbb{R}^{m+1}, & (3.50) \\ q_t &= \left[\begin{array}{ccccc} q_{t,0} & q_{t,1} & \cdots & q_{t,n-1} & q_{t,n} \end{array}\right]^T \in \mathbb{R}^{n+1}, & (3.51) \end{aligned}$$

are vectors of the perturbations that are added to the coefficients of $\tilde{f}_\theta(y)$ and $\tilde{g}_\theta(y)$, respectively. The perturbation β_0 is added to α_0.

The computations of the vectors z_t, p_t, q_t and $\tilde{\mathbf{r}}(\theta)$, and the scalars β_0 and θ require an approximate equation in (3.44) to be solved. This equation is non-linear and the Newton-Raphson method is used to solve this equation.

An approximate solution for (3.44) is associated with the following residual:

$$\begin{aligned} r(\beta_0, \theta, z_t, \tilde{\mathbf{r}}_t, p_t, q_t) &= \left[\begin{array}{c} \tilde{\mathbf{f}}(\theta) + \mathbf{s}_t(p_t, \theta) \\ (\alpha_0 + \beta_0)\left(\tilde{\mathbf{g}}(\theta) + \mathbf{w}_t(q_t, \theta)\right) \end{array}\right] \\ &- \left[\begin{array}{c} C_{t,1}(\tilde{c}_t, \theta) + E_{t,1}(z_t, \theta) \\ C_{t,2}(\tilde{e}_t, \theta) + E_{t,2}(z_t, \theta) \end{array}\right] \tilde{\mathbf{r}}_t(\theta). \qquad (3.52) \end{aligned}$$

Therefore, a first-order Taylor expansion yields

$$\begin{aligned} & r(\beta_0+\delta\beta_0,\theta+\delta\theta,z_t+\delta z_t,\tilde{\mathbf{r}}_t+\delta\tilde{\mathbf{r}}_t,p_t+\delta p_t,q_t+\delta q_t) \\ & = \begin{bmatrix} \tilde{\mathbf{f}}(\theta+\delta\theta)+\mathbf{s}_t(p_t+\delta p_t,\theta+\delta\theta) \\ (\alpha_0+\beta_0+\delta\beta_0)\left(\tilde{\mathbf{g}}(\theta+\delta\theta)+\mathbf{w}_t(q_t+\delta q_t,\theta+\delta\theta)\right) \end{bmatrix} \qquad (3.53) \\ & - \begin{bmatrix} C_{t,1}(\tilde{c}_t,\theta+\delta\theta)+E_{t,1}(z_t+\delta z_t,\theta+\delta\theta) \\ C_{t,2}(\tilde{e}_t,\theta+\delta\theta)+E_{t,2}(z_t+\delta z_t,\theta+\delta\theta) \end{bmatrix} \tilde{\mathbf{r}}_t(\theta+\delta\theta). \qquad (3.54) \end{aligned}$$

where

$$\tilde{\mathbf{r}}_t(\theta+\delta\theta)=\tilde{\mathbf{r}}_t(\theta)+\frac{d(\tilde{\mathbf{r}}_t(\theta))}{d\theta}\delta\theta=\tilde{\mathbf{r}}_t(\theta)+\begin{bmatrix} t\tilde{r}_{t,0}\theta^{t-1} \\ (t-1)\,\tilde{r}_{t,1}\theta^{t-2} \\ \vdots \\ 2\tilde{r}_{t,t-2}\theta \\ \tilde{r}_{t,t-1} \\ 0 \end{bmatrix}. \qquad (3.55)$$

In order to simplify the analysis of the expression of (3.53) and (3.54), let us first consider (3.53).

The approximation of the first order of the first expression in (3.53) is

$$\tilde{\mathbf{f}}(\theta+\delta\theta)+\mathbf{s}_t(p_t+\delta p_t,\theta+\delta\theta) \approx \tilde{\mathbf{f}}+\mathbf{s}_t+\frac{\partial\tilde{\mathbf{f}}}{\partial\theta}\delta\theta+\frac{\partial\mathbf{s}_t}{\partial\theta}\delta\theta+\sum_{i=0}^{m}\frac{\partial\mathbf{s}_t}{\partial p_{t,i}}\delta p_{t,i}, \qquad (3.56)$$

and the approximation of the first order of the second expression is

$$\begin{aligned} & (\alpha_0+\beta_0+\delta\beta_0)\left(\tilde{\mathbf{g}}(\theta+\delta\theta)+\mathbf{w}_t(q_t+\delta q_t,\theta+\delta\theta)\right) \\ & \approx (\alpha_0+\beta_0)(\tilde{\mathbf{g}}+\mathbf{w}_t)+(\alpha_0+\beta_0)\left(\tfrac{\partial\tilde{\mathbf{g}}}{\partial\theta}\delta\theta+\tfrac{\partial\mathbf{w}_t}{\partial\theta}\delta\theta+\textstyle\sum_{i=0}^{n}\tfrac{\partial\mathbf{w}_t}{\partial q_{t,i}}\delta q_{t,i}\right) \qquad (3.57) \\ & +(\tilde{\mathbf{g}}+\mathbf{w}_t)\,\delta\beta_0, \end{aligned}$$

where

$$\frac{\partial \tilde{\mathbf{f}}}{\partial \theta} = \begin{bmatrix} m\bar{a}_0\theta^{m-1} \\ (m-1)\,\bar{a}_1\theta^{m-2} \\ \vdots \\ \bar{a}_{m-1} \\ 0 \end{bmatrix}, \qquad \frac{\partial \mathbf{s}_t}{\partial \theta} = \begin{bmatrix} mp_{t,0}\theta^{m-1} \\ (m-1)\,p_{t,1}\theta^{m-2} \\ \vdots \\ p_{t,m-1} \\ 0 \end{bmatrix}, \tag{3.58}$$

and

$$\frac{\partial \tilde{\mathbf{g}}}{\partial \theta} = \begin{bmatrix} n\bar{b}_0\theta^{n-1} \\ (n-1)\,\bar{b}_1\theta^{n-2} \\ \vdots \\ \bar{b}_{n-1} \\ 0 \end{bmatrix}, \qquad \frac{\partial \mathbf{w}_t}{\partial \theta} = \begin{bmatrix} nq_{t,0}\theta^{n-1} \\ (n-1)\,q_{t,1}\theta^{n-2} \\ \vdots \\ q_{t,n-1} \\ 0 \end{bmatrix}, \tag{3.59}$$

The vectors $\mathbf{s}$ and $\mathbf{w}$ can be written using square diagonal matrices as $\mathbf{s} = Sp_t$ and $\mathbf{w} = Wq_t$, respectively, where

$$S = S(\theta) = \operatorname{diag}\begin{bmatrix} \theta^m & \theta^{m-1} & \dots & \theta & 1 \end{bmatrix} \in \mathbb{R}^{(m+1)\times(m+1)}, \tag{3.60}$$

$$W = W(\theta) = \operatorname{diag}\begin{bmatrix} \theta^n & \theta^{n-1} & \dots & \theta & 1 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}. \tag{3.61}$$

Therefore, it follows that

$$\sum_{i=0}^{m} \frac{\partial \mathbf{s}_t}{\partial p_{t,i}} \delta p_{t,i} = S\delta p_t \qquad \text{and} \qquad \sum_{i=0}^{n} \frac{\partial \mathbf{w}_t}{\partial q_{t,i}} \delta q_{t,i} = W\delta q_t, \tag{3.62}$$

and thus

$$\tilde{\mathbf{f}}(\theta+\delta\theta)+\mathbf{s}_t(p_t+\delta p_t,\theta+\delta\theta) \;\approx\; \tilde{\mathbf{f}}+\mathbf{s}_t+\frac{\partial\tilde{\mathbf{f}}}{\partial\theta}\delta\theta+\frac{\partial\mathbf{s}_t}{\partial\theta}\delta\theta+S\delta p_t, \tag{3.63}$$

and

$$\begin{aligned}&(\alpha_0+\beta_0+\delta\beta_0)\Big(\tilde{\mathbf{g}}(\theta+\delta\theta)+\mathbf{w}_t(q_t+\delta q_t,\theta+\delta\theta)\Big)\\&\approx(\alpha_0+\beta_0)\left(\tilde{\mathbf{g}}+\mathbf{w}_t\right)+(\alpha_0+\beta_0)\left(\frac{\partial\tilde{\mathbf{g}}}{\partial\theta}\delta\theta+\frac{\partial\mathbf{w}_t}{\partial\theta}\delta\theta+W\delta q_t\right)\\&+\left(\tilde{\mathbf{g}}+\mathbf{w}_t\right)\delta\beta_0.\end{aligned} \tag{3.64}$$

Then, by using (3.63) and (3.64), the expression in (3.53) can be written as

$$\begin{bmatrix}\tilde{\mathbf{f}}+\mathbf{s}_t\\ \\(\alpha_0+\beta_0)\left(\tilde{\mathbf{g}}+\mathbf{w}_t\right)\end{bmatrix}+\begin{bmatrix}\frac{\partial\tilde{\mathbf{f}}}{\partial\theta}\delta\theta+\frac{\partial\mathbf{s}_t}{\partial\theta}\delta\theta+S\delta p_t\\ \\(\alpha_0+\beta_0)\left(\frac{\partial\tilde{\mathbf{g}}}{\partial\theta}\delta\theta+\frac{\partial\mathbf{w}_t}{\partial\theta}\delta\theta+W\delta q_t\right)+\left(\tilde{\mathbf{g}}+\mathbf{w}_t\right)\delta\beta_0\end{bmatrix}. \tag{3.65}$$

Now, let us consider the expression in Eqn 3.54. The following equation

$$B_t=B_t(\tilde{c}_t,\tilde{e}_t,\theta)=\begin{bmatrix}C_{t,1}(\tilde{c}_t,\theta)\\ \\C_{t,2}(\tilde{e}_t,\theta)\end{bmatrix}\quad\text{and}\quad E_t=E_t(z_t,\theta)=\begin{bmatrix}E_{t,1}(z_t,\theta)\\ \\E_{t,2}(z_t,\theta)\end{bmatrix}, \tag{3.66}$$

allows the expression in Eqn. 3.54 to be rewritten as

$$-\Big(B_t(\tilde{c}_t,\tilde{e}_t,\theta+\delta\theta)+E_t(z_t+\delta z_t,\theta+\delta\theta)\Big)\tilde{\mathbf{r}}_t(\theta+\delta\theta), \tag{3.67}$$

whose first order equal to

$$\begin{aligned}&-\left(B_t+\frac{\partial B_t}{\partial\theta}\delta\theta+E_t+\frac{\partial E_t}{\partial\theta}\delta\theta+\sum_{i=0}^{m+n-2t+1}\frac{\partial E_t}{\partial z_{t,i}}\delta z_{t,i}\right)\left(\tilde{\mathbf{r}}_t(\theta)+\frac{d\left(\tilde{\mathbf{r}}_t(\theta)\right)}{d\theta}\delta\theta\right)\\=\;&-\left(B_t+E_t\right)\tilde{\mathbf{r}}_t-\left(\frac{\partial B_t}{\partial\theta}\delta\theta+\frac{\partial E_t}{\partial\theta}\delta\theta+\delta E_t\right)\tilde{\mathbf{r}}_t-\left(B_t+E_t\right)\frac{d\tilde{\mathbf{r}}_t}{d\theta}\delta\theta,\end{aligned} \tag{3.68}$$

where

$$\frac{\partial B_t}{\partial \theta} = \begin{bmatrix} \frac{\partial C_{t,1}(\tilde{c}_t,\theta)}{\partial \theta} \\ \frac{\partial C_{t,2}(\tilde{e}_t,\theta)}{\partial \theta} \end{bmatrix} \quad \text{and} \quad \delta E_t = \sum_{i=0}^{m+n-2t+1} \frac{\partial E_t}{\partial z_{t,i}} \delta z_{t,i}. \tag{3.69}$$

can be used to obtain the matrices $\frac{\partial C_{t,1}(\tilde{c}_t,\theta)}{\partial \theta}$ and $\frac{\partial C_{t,2}(\tilde{e}_t,\theta)}{\partial \theta}$ from (3.41) and (3.42), respectively,

$$\begin{bmatrix} (m-t)\,\tilde{c}_{t,0}\theta^{m-t-1} & & & \\ (m-t-1)\,\tilde{c}_{t,1}\theta^{m-t-2} & (m-t)\,\tilde{c}_{t,0}\theta^{m-t-1} & & \\ (m-t-2)\,\tilde{c}_{t,2}\theta^{m-t-3} & (m-t-1)\,\tilde{c}_{t,1}\theta^{m-t-2} & \ddots & \\ \vdots & (m-t-2)\,\tilde{c}_{t,2}\theta^{m-t-3} & \ddots & (m-t)\,\tilde{c}_{t,0}\theta^{m-t-1} \\ \vdots & \vdots & \ddots & (m-t-1)\,\tilde{c}_{t,1}\theta^{m-t-2} \\ \tilde{c}_{t,m-t-1} & \vdots & \ddots & (m-t-2)\,\tilde{c}_{t,2}\theta^{m-t-3} \\ 0 & \tilde{c}_{t,m-t-1} & \ddots & \vdots \\ & 0 & \ddots & \vdots \\ & & \ddots & \tilde{c}_{t,m-t-1} \\ & & & 0 \end{bmatrix} \tag{3.70}$$

and

$$\left[\begin{array}{cccc}
(n-t)\,\tilde{e}_{t,0}\theta^{n-t-1} & & & \\
(n-t-1)\,\tilde{e}_{t,1}\theta^{n-t-2} & (n-t)\,\tilde{e}_{t,0}\theta^{n-t-1} & & \\
(n-t-2)\,\tilde{e}_{t,2}\theta^{n-t-3} & (n-t-1)\,\tilde{e}_{t,1}\theta^{n-t-2} & \ddots & \\
\vdots & (n-t-2)\,\tilde{e}_{t,2}\theta^{n-t-3} & \ddots & (n-t)\,\tilde{e}_{t,0}\theta^{n-t-1} \\
\vdots & \vdots & \ddots & (n-t-1)\,\tilde{e}_{t,1}\theta^{n-t-2} \\
\tilde{e}_{t,n-t-1} & \vdots & \ddots & (n-t-2)\,\tilde{e}_{t,2}\theta^{n-t-3} \\
0 & \tilde{e}_{t,n-t-1} & \ddots & \vdots \\
 & 0 & \ddots & \vdots \\
 & & \ddots & \tilde{e}_{t,n-t-1} \\
 & & & 0
\end{array}\right]. \tag{3.71}$$

Moreover, let $Y_t(\tilde{\mathbf{r}}_t, \theta) \in \mathbb{R}^{(m+n+2)\times(m+n-2t+2)}$, and

$$Y_t(\tilde{\mathbf{r}}_t, \theta) = \begin{bmatrix} Y_{t,1}(\tilde{\mathbf{r}}_t, \theta) \\ Y_{t,2}(\tilde{\mathbf{r}}_t, \theta) \end{bmatrix}, \tag{3.72}$$

where $Y_{t,1}(\tilde{\mathbf{r}}_t, \theta) \in \mathbb{R}^{(m+1)\times(m+n-2t+2)}$ and $Y_{t,2}(\tilde{\mathbf{r}}_t, \theta) \in \mathbb{R}^{(n+1)\times(m+n-2t+2)}$,

$$Y_{t,1}(\tilde{\mathbf{r}}_t, \theta) \quad = \quad \begin{bmatrix} C_{t,3}(\tilde{\mathbf{r}}_t)\Theta_{t,1} & 0_{m+1,n-t+1} \end{bmatrix}, \quad C_{t,3}(\tilde{\mathbf{r}}_t) \in \mathbb{R}^{(m+1)\times(m-t+1)}, \tag{3.73}$$

$$Y_{t,2}(\tilde{\mathbf{r}}_t, \theta) \quad = \quad \begin{bmatrix} 0_{n+1,m-t+1} & C_{t,4}(\tilde{\mathbf{r}}_t)\Theta_{t,2} \end{bmatrix}, \quad C_{t,4}(\tilde{\mathbf{r}}_t) \in \mathbb{R}^{(n+1)\times(n-t+1)}, \tag{3.74}$$

where $C_{t,3}(\tilde{\mathbf{r}}_t)$ and $C_{t,4}(\tilde{\mathbf{r}}_t)$ have the same form, which are Toeplitz matrices of $\tilde{\mathbf{r}}_t$, but with different dimensions, and $\Theta_{t,1}$ and $\Theta_{t,2}$ are given by

$$\Theta_{t,1} = \operatorname{diag}\left[\begin{array}{ccccc} \theta^{m-t} & \theta^{m-t-1} & \dots & \theta & 1 \end{array}\right] \in \mathbb{R}^{(m-t+1)\times(m-t+1)}, \tag{3.75}$$

$$\Theta_{t,2} = \operatorname{diag}\left[\begin{array}{ccccc} \theta^{n-t} & \theta^{n-t-1} & \dots & \theta & 1 \end{array}\right] \in \mathbb{R}^{(n-t+1)\times(n-t+1)}. \tag{3.76}$$

The expression in (3.68) can be simplified by differentiating, with respect to z, both sides of the equation.

$$Y_t(\tilde{\mathbf{r}}_t, \theta) z_t = E_t(z_t, \theta)\tilde{\mathbf{r}}_t, \tag{3.77}$$

to give,

$$\delta E_t(z_t, \theta)\tilde{\mathbf{r}}_t = Y_t(\tilde{\mathbf{r}}_t, \theta)\delta z_t. \tag{3.78}$$

It follows that, (3.54) can be written as

$$-\left(B_t + E_t\right)\tilde{\mathbf{r}}_t - \left(B_t + E_t\right)\frac{d\tilde{\mathbf{r}}_t}{d\theta}\delta\theta - Y_t\delta z_t - \left(\frac{\partial B_t}{\partial\theta}\tilde{\mathbf{r}} + \frac{\partial E_t}{\partial\theta}\tilde{\mathbf{r}}_t\right)\delta\theta. \tag{3.79}$$

Then, the expressions in (3.65) and (3.79) are substituted to (3.53) and (3.54), respectively, yielding

$$\begin{aligned} & r(\beta_0 + \delta\beta_0, \theta + \delta\theta, z_t + \delta z_t, \tilde{\mathbf{r}}_t + \delta\tilde{\mathbf{r}}_t, p_t + \delta p_t, q_t + \delta q_t) \\ & \approx r(\beta_0, \theta, z_t, \tilde{\mathbf{r}}_t, p_t, q_t) \\ & + \left[\begin{array}{cccc} S & 0_{m+1,n+1} & 0_{m+1,1} & \frac{\partial\tilde{\mathbf{f}}_t}{\partial\theta} + \frac{\partial\mathbf{s}_t}{\partial\theta} \\ 0_{n+1,m+1} & \left(\alpha_0 + \beta_0\right)W & \tilde{\mathbf{g}}_t + \mathbf{w}_t & \left(\alpha_0 + \beta_0\right)\left(\frac{\partial\tilde{\mathbf{g}}_t}{\partial\theta} + \frac{\partial\mathbf{w}_t}{\partial\theta}\right) \end{array}\right]\left[\begin{array}{c} \delta p_t \\ \delta q_t \\ \delta\beta_0 \\ \delta\theta \end{array}\right] \\ & - \left(B_t + E_t\right)\frac{d\tilde{\mathbf{r}}_t}{d\theta}\delta\theta - Y_t\delta z_t - \left(\frac{\partial B_t}{\partial\theta}\tilde{\mathbf{r}}_t + \frac{\partial E_t}{\partial\theta}\tilde{\mathbf{r}}_t\right)\delta\theta. \end{aligned} \tag{3.80}$$

Therefore, the calculation of the jth iteration in the Newton-Raphson method for the solution of z, p, q, β_0 and θ is

$$\left[\begin{array}{c|cccc} Y_t & -S & 0_{m+1,n+1} & 0_{m+1,1} & -\left(\frac{\partial \tilde{\mathbf{f}}}{\partial \theta}+\frac{\partial \mathbf{s}_t}{\partial \theta}\right)+\left(\frac{\partial C_{t,1}}{\partial \theta}+\frac{\partial E_{t,1}}{\partial \theta}\right)\tilde{\mathbf{r}}_t+\left(C_{t,1}+E_{t,1}\right)\frac{d\tilde{\mathbf{r}}_t}{d\theta} \\ & 0_{n+1,m+1} & -\left(\alpha_0+\beta_0\right)W & -(\tilde{\mathbf{g}}+\mathbf{w}_t) & -(\alpha_0+\beta_0)\left(\frac{\partial \tilde{\mathbf{g}}}{\partial \theta}+\frac{\partial \mathbf{w}_t}{\partial \theta}\right)+\left(\frac{\partial C_{t,2}}{\partial \theta}+\frac{\partial E_{t,2}}{\partial \theta}\right)\tilde{\mathbf{r}}_t+\left(C_{t,2}+E_{t,2}\right)\frac{d\tilde{\mathbf{r}}_t}{d\theta} \end{array}\right]^{(j)} \begin{bmatrix} \delta z_t \\ \delta p_t \\ \delta q_t \\ \delta \beta_0 \\ \delta \theta \end{bmatrix}^{(j)}$$

$$= r^{(j)}(\beta_0, \theta, z_t, \tilde{\mathbf{r}}_t, p_t, q_t). \tag{3.81}$$

In the $(j+1)$th iteration, the improved estimates of z_t, p_t, q_t, β_0 and θ are obtained from

$$\begin{bmatrix} z_t \\ p_t \\ q_t \\ \beta_0 \\ \theta \end{bmatrix}^{(j+1)} = \begin{bmatrix} z_t \\ p_t \\ q_t \\ \beta_0 \\ \theta \end{bmatrix}^{(j)} + \begin{bmatrix} \delta z_t \\ \delta p_t \\ \delta q_t \\ \delta \beta_0 \\ \delta \theta \end{bmatrix}^{(j)}, \tag{3.82}$$

where the initial values are

$$z_t^{(0)} = 0, \qquad p_t^{(0)} = 0, \qquad q_t^{(0)} = 0, \qquad \beta_0^{(0)} = 0, \qquad \theta^{(0)} = \theta_0. \tag{3.83}$$

Furthermore, the initial value of $\tilde{\mathbf{r}}_t(\theta)$ is estimated by the least squares solution of (3.40)

$$\tilde{\mathbf{r}}_t^{(0)}(\theta_0) \approx \begin{bmatrix} C_{t,1}(\tilde{c}_t, \theta_0) \\ C_{t,2}(\tilde{e}_t, \theta_0) \end{bmatrix}^{\dagger} \begin{bmatrix} \tilde{\mathbf{f}}(\theta_0) \\ \alpha_0 \tilde{\mathbf{g}}(\theta_0) \end{bmatrix}. \tag{3.84}$$

where $X^{\dagger} = (X^T X)^{-1} X^T$.

It is clearly obvious that (3.81) is of the form

$$C^{(j)} y^{(j)} = g^{(j)}, \tag{3.85}$$

$$C^{(j)} = \left[\begin{array}{c|cccc} Y_t & -S & 0_{m+1,n+1} & 0_{m+1,1} & -\left(\frac{\partial \tilde{\mathbf{f}}}{\partial \theta} + \frac{\partial \mathbf{s}_t}{\partial \theta}\right) + \left(\frac{\partial C_{t,1}}{\partial \theta} + \frac{\partial E_{t,1}}{\partial \theta}\right) \tilde{\mathbf{r}}_t + (C_{t,1} + E_{t,1}) \frac{d\tilde{\mathbf{r}}_t}{d\theta} \\ & 0_{n+1,m+1} & -(\alpha_0 + \beta_0) W & -(\tilde{\mathbf{g}} + \mathbf{w}_t) & -(\alpha_0 + \beta_0) \left(\frac{\partial \tilde{\mathbf{g}}}{\partial \theta} + \frac{\partial \mathbf{w}_t}{\partial \theta}\right) + \left(\frac{\partial C_{t,2}}{\partial \theta} + \frac{\partial E_{t,2}}{\partial \theta}\right) \tilde{\mathbf{r}}_t + (C_{t,2} + E_{t,2}) \frac{d\tilde{\mathbf{r}}_t}{d\theta} \end{array} \right]^{(j)}, \tag{3.86}$$

where $C^{(j)} \in \mathbb{R}^{(m+n+2)\times(2m+2n-2t+6)}$, $y^{(j)} \in \mathbb{R}^{2m+2n-2t+6}$, $g^{(j)} \in \mathbb{R}^{m+n+2}$,

$$y^{(j)} = \begin{bmatrix} \delta z_t^{(j)} \\ \delta p_t^{(j)} \\ \delta q_t^{(j)} \\ \delta \beta_0^{(j)} \\ \delta \theta^{(j)} \end{bmatrix} \quad \text{and} \quad g^{(j)} = r^{(j)}(\beta_0, \theta, z_t, \tilde{\mathbf{r}}_t, p_t, q_t). \tag{3.87}$$

Since a requirement is to calculate the solution that is nearest to the given inexact data, it therefore must minimise the function

$$\left\| \begin{bmatrix} z_t^{(j+1)} - zt^{(0)} \\ p_t^{(j+1)} - pt^{(0)} \\ qt^{(j+1)} - qt^{(0)} \\ \beta_0^{(j+1)} - \beta_0^{(0)} \\ \theta^{(j+1)} - \theta_0 \end{bmatrix} \right\| = \left\| \begin{bmatrix} z_t^{(j)} + \delta z_t^{(j)} \\ p_t^{(j)} + \delta p_t^{(j)} \\ q_t^{(j)} + \delta q_t^{(j)} \\ \beta_0^{(j)} + \delta \beta_0^{(j)} \\ \theta^{(j)} + \delta \theta^{(j)} - \theta_0 \end{bmatrix} \right\| := \left\| E^{(j)} y^{(j)} - f^{(j)} \right\|, \tag{3.88}$$

where

$$E^{(j)} = I_{2m+2n-2t+6}, \quad \text{and} \quad f^{(j)} = -\left[\begin{array}{ccccc} z_t^{(j)} & p_t^{(j)} & q_t^{(j)} & \beta_0^{(j)} & \theta^{(j)} - \theta_0 \end{array}\right]^T. \tag{3.89}$$

Consequently, the method of SNTLN yields the following least squares problem (LSE) with an equality constraint.

$$\min_{y^{(j)}} \left\| E^{(j)} y^{(j)} - f^{(j)} \right\| \quad \text{subject to} \quad C^{(j)} y^{(j)} = g^{(j)}. \tag{3.90}$$

$C^{(j)}, f^{(j)}$, and $g^{(j)}$ are updated between successive iterations, where the initial value of $f = 0$. In each iteration, the vector y will be calculated and this enables f to be updated from (3.68).

To conclude, the investigation of this method has shown that good results can be achieved with polynomials that have multiple roots of high degree.

3.6 Summary

This chapter started with a review of the basic concepts of the GCD of two polynomials, including the definition and its computation using the Sylvester resultant matrix. Then, the computation of AGCD that is applied in this research was discussed, along with the three pre-processing operations that must be applied to improve the reliability of the computations performed on the Sylvester resultant matrix. The computation of an AGCD of two inexact polynomial is separated into two stages. Firstly, calculation of the degree of an AGCD and it is done using the QR decomposition. Then, the coefficient of an AGCD is computed explicitly since applying the APF. The final section of the chapter considered the method of APF in detail. Due to the ill-posed problem, in the literature, many of the algorithms of AGCD fail to compute the correct value of its degree and coefficient [80]. The algorithm used in this study has been used to compute the multiple roots of hard classes of inexact polynomials and it is shown that it gives very good results [80]. In addition to the high accuracy of AGCD algorithm that is described in this chapter, there are some advantages in using the APF and QR decomposition. First, it can compute the numerical rank of the Sylvester resultant matrix directly using the given data without any prior information. Second, the multiplicity structure of the polynomial is preserved. Third, exploitation of the properties of QR decomposition makes the developed algorithm much faster that SVD. Finally, the APF implementation for the AGCD computation using SNTLN results in explicitly computed coefficients without the need for an extra computation stage as another method. However, a disadvantage of this method is that the given data must be non-zero value as the geometric mean of the Sylvester resultant matrix is applied. This case is not considered an issue in practical cases as its occurrence is not very likely to happen.

Chapter 4

Blind Image Deconvolution for Separable PSF

4.1 Introduction

The limitations of the literature regarding the solution of the BID problem were discussed in Chapter 2. This chapter considers the application of a solution of BID using linear algebra, specifically, operations on polynomials. The pixel values of the blurred image $\mathcal{G}$, deblurred image $\mathcal{F}$, and the PSF $\mathcal{H}$ can be considered as coefficients of bivariate polynomials, and G is the multiplication of two bivariate polynomials F and H, in which the polynomials are the representation G, F, and H of $\mathcal{G}$, $\mathcal{F}$, and $\mathcal{H}$, respectively. Therefore, this consideration allows $\mathcal{H}$ to be computed as the Approximate Greatest Common Divisor (AGCD), which is then deconvolved from the blurred image $\mathcal{G}$, thereby obtaining a good approximation of the original image $\mathcal{F}$. The AGCD computations are performed using an approximate factorisation of two inexact polynomials that is used to calculate a structured low-rank approximation of the Sylvester resultant matrix of two inexact polynomials. As was mentioned in Chapter 2, the theory of linear algebra has also been used for BID in [18, 19, 38]. However, the work developed in this research differs from the work in these papers in several ways:

- This work allows the degrees and coefficients of the AGCD, which consist of the horizontal and vertical extents of the PSF, to be calculated without any knowledge about the noise level or the signal-to-noise ratio (SNR). This is unlike the work reported in [1, 2, 38] which required the SNR to be known for the calculation of the degrees of the AGCD. The SNR is the threshold in a stopping criterion in the algorithm. In contrast, [18, 19] used performance and visual inspection of the deblurred images to determine estimates of the degrees of the AGCD.
- This work is applicable for computing the PSF of various and wide width sizes in opposition to that reported in [38].
- The Fourier transform is used to calculate a separable PSF in [1, 2, 18, 19, 38]. In contrast, the developed work uses the exact pixel values of blurred and deblurred images and the PSF as the coefficients of polynomials, and it shows that the Fourier transform is not required. However, the developed work uses the Fourier transform and non-linear structured matrix method for a non-separable PSF in order to compare the result with the previous work in the literature [1, 2](see Chapter 6).

The developed method including the calculation of the PSF and its deconvolution from a blurred image is considered in this chapter, starting in Section 4.2 by representing the convolution operation that defines the formation of G and the multiplication of two polynomials F and H. It is assumed that the PSF is separable and Section 4.3 interprets the separable PSF that is equal to an AGCD of two polynomials. The extension of the method to form a non-separable PSF from a separable PSF is discussed in the next chapter.

4.2 The Convolution Operation

The mathematical model of an image represents the image as a two-dimensional, in which its components are the pixel values of the image. The method discussed in this thesis works with a grayscale image that can be represented as a matrix. The matrix represents the image as bivariate polynomials and its coefficients are the components of the matrix. In particular, let the coefficients $f(i,j)$ of the bivariate polynomial $F(x,y)$ be the pixel value of the original image $\mathcal{F}$, and $F(x,y)$ be of the degrees $M-1$ in x and $N-1$ in y,

$$F(x,y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) x^{M-1-i} y^{N-1-j}, \tag{4.1}$$

In addition, let the coefficients $h(k,l)$ of a bivariate polynomial $H(x,y)$ be the pixel value of a spatially invariant PSF $\mathcal{H}$, and $H(x,y)$ be of the degrees p in x and r in y,

$$H(x,y) = \sum_{k=0}^{p} \sum_{l=0}^{r} h(k,l) x^{p-k} y^{r-l}. \tag{4.2}$$

Consequently, the convolution in (2.1) reduces to the multiplication of two polynomials. In the absence of noise, the blurred image is formed by the multiplication of the original image $\mathcal{F}$ with PSF $\mathcal{H}$,

$$G(x,y) = F(x,y)H(x,y), \tag{4.3}$$

$$G(x,y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=0}^{p} \sum_{l=0}^{r} f(i,j)h(k,l) x^{M+p-1-(i+k)} y^{N+r-1-(j+l)}, \tag{4.4}$$

and substituting $(i+k)$ with s, and $(j+l)$ with t, yields

$$G(x,y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{s=i}^{p+i} \sum_{t=j}^{r+j} f(i,j)h(s-i,t-j)x^{M+p-1-s}y^{N+r-1-t}. \tag{4.5}$$

Thereby the coefficient of $x^{M+p-1-s}y^{N+r-1-t}$ in $G(x,y)$ is

$$g(s,t) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j)h(s-i,t-j), \tag{4.6}$$

where $h(k,l) = 0$ if $k < 0$ or $l < 0$. It follows that equation (2.3) is equivalent to (4.6), which means that image $\mathcal{F}$ with $\mathcal{H}$ is equivalent to the multiplication of the polynomial form of the exact image $\mathcal{F}$ by the polynomial form of $\mathcal{H}$. Therefore, the deblurred image can be computed by the division of the polynomial form of $\mathcal{G}$ by the polynomial form of $\mathcal{H}$.

Equation (4.5) and (4.6) show that the blurred image $\mathcal{G}$ is larger than the original image $\mathcal{F}$. Specifically, if F is a polynomial of degree $M-1$ in x and $N-1$ in y, where x and y are the columns and rows of the matrix representing the image, and H is a polynomial of degrees p and r in x and y, respectively, it therefore follows that G is a polynomial of degrees $M+p-1$ in x and $N+r-1$ in y, and thus the extension of pixels along the columns and rows from F to G is p and r. This extension is removed when the H is deconvolved from G, thereby yielding a deblurred image which will be of the same size as F.

4.3 Polynomial Computations for BID

Section 4.2 showed that an image that was blurred by a spatially invariant blur can be produced by the multiplication of the bivariate polynomials of the exact image with a PSF. This multiplication of polynomials shows that the PSF is equal to an AGCD of two polynomials. This section considers the polynomial multiplication in detail. It follows from section 4.2 that, in noise free cases, equation (2.1) can be written as

$$G(x, y) = H(x, y)F(x, y), \tag{4.7}$$

where $H(x, y)$ is a separable PSF that satisfies

$$H(x, y) = H_c(x)H_r(y). \tag{4.8}$$

then (4.7) is generalised to

$$G(x, y) = H_c(x)H_r(y)F(x, y). \tag{4.9}$$

Consider two rows and two columns of $\mathcal{G}$, $x = r_1$ and $x = r_2$, and $y = c_1$ and $y = c_2$, respectively,

$$G(r_1, y) = H_c(r_1)H_r(y)F(r_1, y), \tag{4.10}$$

$$G(r_2, y) = H_c(r_2)H_r(y)F(r_2, y), \tag{4.11}$$

$$G(x, c_1) = H_c(x)H_r(c_1)F(x, c_1), \tag{4.12}$$

$$G(x, c_2) = H_c(x)H_r(c_2)F(x, c_2), \tag{4.13}$$

where $H_c(r_1), H_c(r_2), H_r(c_1)$ and $H_r(c_2)$ are constants. Since the equations for the rows r_1 and r_2 have the same form as the equations for the columns c_1 and c_2, it is adequate to consider one of them only, either the equations for the rows or the columns. Therefore, consider the equations for the rows which are in the independent variable y,

$$G(r_1, y) = H_c(r_1)H_r(y)F(r_1, y), \tag{4.14}$$

$$G(r_2, y) = H_c(r_2)H_r(y)F(r_2, y). \tag{4.15}$$

It is obvious that if the polynomials form of the rows $F(r_1, y)$ and $F(r_2, y)$, of $\mathcal{F}$, are coprime, then

$$H_r(y) = \text{AGCD}\Big(G(r_1, y), G(r_2, y)\Big). \tag{4.16}$$

Because of the assumption that the PSF is separable, the arguments r_1 and r_2 appear on the right-hand side, not on the left-hand side of this equation, and they are indices to any two rows of $\mathcal{G}$. Equation (4.16) shows that if we consider the pixel values of a given blurred image as the coefficient of a polynomial, the row component $H_r(y)$ of a spatially invariant and separable PSF is equal to an AGCD of any two rows of the blurred image. It is clear that the computation of the column component of the PSF follows identically. Then, the PSF can be computed from (4.8). After that, the deblurred image is calculated as divisions of two approximate polynomials.

$$\hat{F}(x, y) \approx \frac{G(x, y)}{H_c(x)H_r(y)}. \tag{4.17}$$

The approximate polynomial divisions are done separately, starting with the row component $H_r(y)$ of the computed PSF

$$Q(x, y) \approx \frac{G(x, y)}{H_r(y)}, \tag{4.18}$$

where $Q(x, y)$ represents the partially deblurred image $\mathcal{Q}$ in polynomial form after the $H_r(y)$ has been deconvolved from $G(x, y)$. It follow that

$$H_r(y)Q(x, y) \approx G(x, y), \tag{4.19}$$

and this approximation equation is applied to all the rows of $\mathcal{G}$. In matrix form, the approximation equation (4.19) can be written as

$$T_r q_i \approx g_i, \qquad i = 1, \ldots, M + p, \tag{4.20}$$

where $T_r \in \mathbb{R}^{(N+r)\times N}$ is a lower triangular Toeplitz matrix, whose entries are the coefficients of $H_r(y)$, $q_i \in \mathbb{R}^N$ is the conversion of the ith row of $\mathcal{Q}$, and $g_i \in \mathbb{R}^{N+r}$ is the conversion of the ith row of the $\mathcal{G}$. Moreover, r and p are defined in (4.2). The blurred image $\mathcal{Q}$ is obtained by solving (4.20) in the least squares sense for each vector q_i. Then, in order to obtain the deblurred image $\hat{\mathcal{F}}$, the column component of the PSF must be deconvolved from $\mathcal{Q}$ and it follows from (4.17) and (4.18) that

$$H_c(x)\hat{F}(x,y) \approx Q(x,y), \tag{4.21}$$

which is also applied to all the columns of $\mathcal{Q}$. Identically with (4.20), the N approximations in a matrix form can be written as follows:

$$T_c\hat{f}_j \approx q_j, \qquad j = 1,\ldots,N, \tag{4.22}$$

where $T_c \in \mathbb{R}^{(M+p)\times M}$ is a Toeplitz matrix, whose entries are the coefficients of $H_c(x)$, $\hat{f}_j \in \mathbb{R}^M$ is the jth column of $\hat{\mathcal{F}}$, and $q_j \in \mathbb{R}^{M+p}$ is the jth column of $\mathcal{Q}$.

Consequently, the deblurred image $\hat{\mathcal{F}}$ that is defined in (4.17) is formed from the vectors $\hat{f}_j$.

$$\hat{F} = [\hat{f}_1\hat{f}_2\ldots\hat{f}_{N-1}\hat{f}_N] \in \mathbb{R}^M. \tag{4.23}$$

The analysis in this section shows that the PSF can be obtained by two separate AGCD computations. These operations are ill-posed and thus the AGCD computations must be done with care. Therefore, the AGCD computations are performed using an approximate polynomial factorisation by the Sylvester resultant matrix of two inexact polynomials that was discussed in Chapter 3.

4.4 Implementations

The implementation of the BID developed method for separable PSF is considered in this section. The algorithm has been implemented in the MATLAB programming environment. This environment provides an excellent framework for dealing with matrix computation.

The developed BID procedure for separable PSF starts with loading the raw blurred image into MATLAB using a standard MATLAB function. Given the PSF is assumed to be separable and spatially invariant, two columns and two rows are needed from the blurred image to calculate the size of the PSF matrix. Twenty five pairs of rows and twenty five pairs of columns are selected randomly for the computation of the size of PSF. Each of the twenty five pairs generates twenty five sizes for the PSF matrix rows and columns. The PSF matrix size is then selected on the basis of the most common (mode) size found for the rows and most common size for columns. The PSF sizes are represented by the degree of the AGCD. The details of the computation of the degree of the AGCD are found in section 3.5.1.

The mode is chosen as the method for selecting the size of the PSF matrix since it is the computed degree for a pair of rows or columns and not an average of the computed degrees. The median can also be used since it is again a computed degree. The mean is deemed to be not suitable for the computation of the degree of the PSF matrix, since it is an average of the computed degrees. The performed experiment showed that the mode and median were the same; however, the mean showed different values from the mode or median.

The coefficients of the PSF matrix of size as determined above is then computed from a randomly selected pair of rows and columns from the blurred image. The deblurred image is then deconvolved using the computed PSF and the blurred image. Algorithm 4.4 gives the MATLAB implementation of the Blind Image Deconvolution for separable and spatially invariant blur.

Algorithm 4.4: Developed Blind Image Deconvolution for Separable and Spatially Invariant PSF

Input: A blurred image $\mathcal{G}$.

Output: A deblurred image $\hat{\mathcal{F}}$ and computed PSF $\hat{\mathcal{H}}$.

Begin:

1. Calculate the size of $\mathcal{G}$, which is $M \times N$.

   % Calculate the size of the PSF.

   % Twenty five trials are used in the computation of the size of the PSF represented by the degree of AGCD

2. Define two random vectors $RR1$ and $RR2$ of size twenty five for rows random selection with values ranging from one to maximum size of $\mathcal{G}$ row.

3. Define two random vectors $RC1$ and $RC2$ of size twenty five for columns random selection with values ranging from one to maximum size of $\mathcal{G}$ column.

4. Initialise two vectors of size twenty five with zeros DR and DC to store the result of the calculated degrees of the AGCD.

5. **while** $i \leq 25$ **do**

6. $r1 = \mathcal{G}(RR1(i), :)$;

7. $r2 = \mathcal{G}(RR2(i), :)$;

8. Pre-process $r1$ and $r2$ using the algorithms detailed in section 3.4 which produces $\tilde{r1}_{\theta_0}$ and $\alpha_0 \tilde{r2}_{\theta_0}$.

9. Calculate the degree of AGCD$(\tilde{r1}_{\theta_0}, \alpha_0 \tilde{r2}_{\theta_0})$ using the QR decomposition of the Sylvester resultant matrix and its sub-resultant matrix $S(\tilde{r1}_{\theta_0}, \alpha_0 \tilde{r2}_{\theta_0})$ as detailed in section 3.5.1.

```
    Save the result in $DR(i)$.
end
while $i \leq 25$ do
    $c1 = \mathcal{G}(:, RC1(i));$
    $c2 = \mathcal{G}(:, RC2(i));$
    Pre-process $c1$ and $c2$ using the algorithms detailed in section 3.4 which produces $\tilde{c1}_{\theta_0}$ and $\alpha_0\tilde{c2}_{\theta_0}$.
    Calculate the degree of AGCD$(\tilde{c1}_{\theta_0}, \alpha_0\tilde{c2}_{\theta_0})$ using the $QR$ decomposition of the Sylvester resultant matrix and its sub-resultant matrix $S(\tilde{c1}_{\theta_0}, \alpha_0\tilde{c2}_{\theta_0})$ as detailed in section 3.5.1.
    Save the result in $DC(i)$.
end
Set $Degree_R = mode(DR);$
Set $Degree_C = mode(DC);$
% Calculate the entries of the PSF matrix represented by the coefficients of AGCD using $Degree_R$ and $Degree_C$.
Set $r1 = \mathcal{G}(min(RR1), :);$
Set $r2 = \mathcal{G}(max(RR2), :);$
Preprocess $r1$ and $r2$ resulting in $\tilde{r1}_{\theta_0}$ and $\alpha_0\tilde{r2}_{\theta_0}$ using the algorithms detailed in section 3.4.
Compute the coefficients of AGCD$(\tilde{r1}_{\theta_0}, \alpha_0\tilde{r2}_{\theta_0})$ results in $HR$ as detailed in section 3.5.2.
Set $c1 = \mathcal{G}(:, min(RC1));$
Set $c2 = \mathcal{G}(:, max(RC2));$
```

```
27. Preprocess $c1$ and $c2$ resulting in $\tilde{c1}_{\theta_0}$ and $\alpha_0\tilde{c2}_{\theta_0}$ using the algorithms detailed in section 3.4.
28. Compute the coefficients of AGCD$(\tilde{c1}_{\theta_0}, \alpha_0\tilde{c2}_{\theta_0})$ results in $HC$ as detailed in section 3.5.2.
29. Form the PSF matrix as a product of $HR$ and $HC$.
30. $\hat{\mathcal{H}} = HR.HC$
    % Compute the deblurred image $\hat{\mathcal{F}}$.
31. $\hat{\mathcal{F}} = \frac{\mathcal{G}}{\hat{\mathcal{H}}}$.
End
```

Algorithm 4.4 is used in the following examples to deblur raw images produced from exact images. The blurred raw image is produced using the following formula:

$$\mathcal{G} = (\mathcal{H} + \mathcal{E}) \otimes \mathcal{F} + \mathcal{N} \tag{4.24}$$

The above blurring formula contains noise terms $\mathcal{E}$ and $\mathcal{N}$. However, in this research study noise was not addressed and has been left for future investigations. In the following experiments very low and insignificant white noise levels has been added. These levels are impractical and cannot be used to justify the developed algorithms performance with noise.

4.4.1 Example

This section contains three examples in which the distorted image is restored using Algorithm 4.4. Raw images of 180 × 180 pixels have been used to examine the developed method performance in recovering the true image and the associated PSF. The following examples show that the developed method resulted in a better image estimation performance compared with other methods described in Chapter 2. This better performance can be attributed to the preservation of the polynomial representation of the image. The Teoplitz structure matrices that represent the images are preserved throughout the algorithm derivation.

Example 4.1. *Grass true image of size* 180×180 *pixels processed with a Gaussian blurring PSF of size* 9×9.

The distorted image produced using the above PSF and recovered image using Algorithm 4.4 are shown in Figure 4.3. Algorithm 4.4

The results of the developed AGCD method processing are given in Figures 4.1, 4.2 and 4.3. Figure 4.1 shows the distribution of the computed PSF row and column sizes. The most frequently computed rank (the mode) is 8 *for rows and columns, and therefore the size of the PSF is* 9×9. *Table 4.1 shows the mode, median and mean values for the distribution of the computed rank for both rows and columns. The table shows that the mode, median and mean are the same since all the computed rank values are the same.*

Figure 4.2 shows the computed PSF in comparison with the exact PSF, and it shows clearly that the developed method gives a good estimate of the PSF.

Figure 4.3 shows the restored image using the developed method. Figure 4.4 and Table 4.2 gives a comparison of the SNR, MSE and Relative error performance measures among the state-of-the-art BID methods. PSNR was not used since it did not show any significant differences among all the compared methods.

Table 4.1: PSF's computed degrees for Example 4.1.

Computing Rank	Mode	Median	Mean
Rows	8	8	8
Columns	8	8	8

Table 4.2 shows that the developed method has better estimation performance in terms of SNR, MSE and Relative error despite that the algorithm assumed no prior information of the PSF or its size.

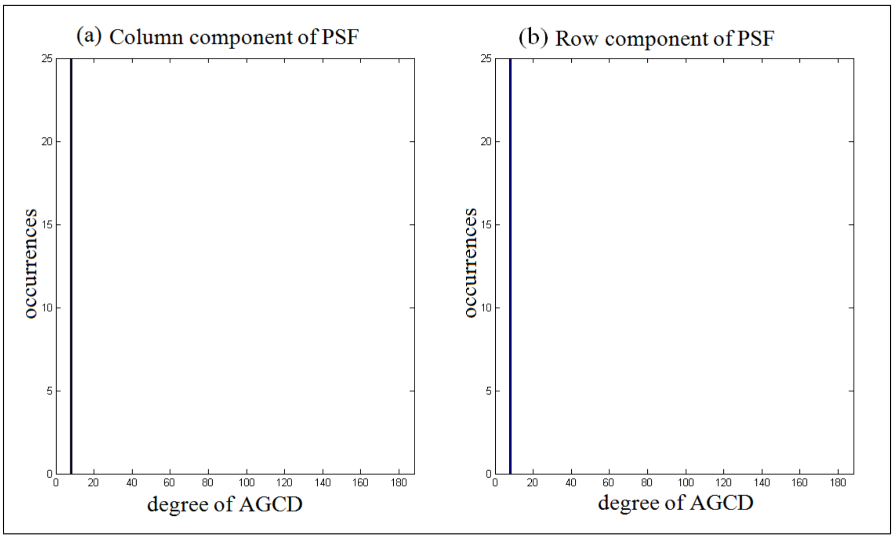


FIGURE 4.1: PSF's computed row and column size using the developed AGCD method for Example 4.1.

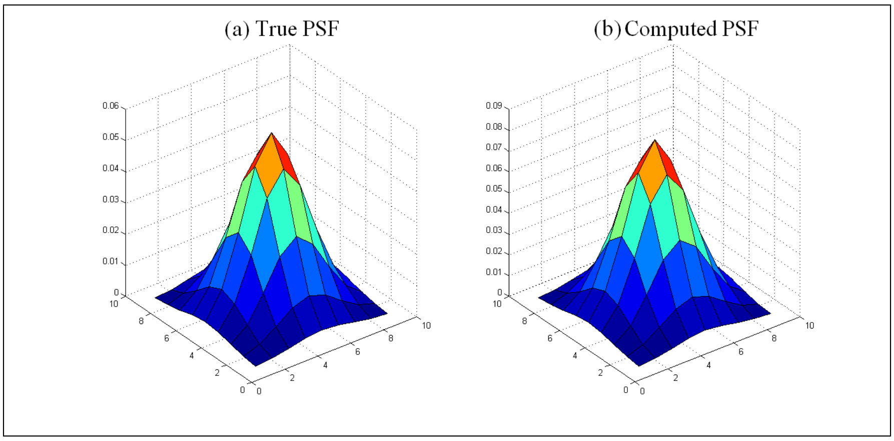


FIGURE 4.2: The true and computed PSFs that are applied to the distorted image in Example 4.1.

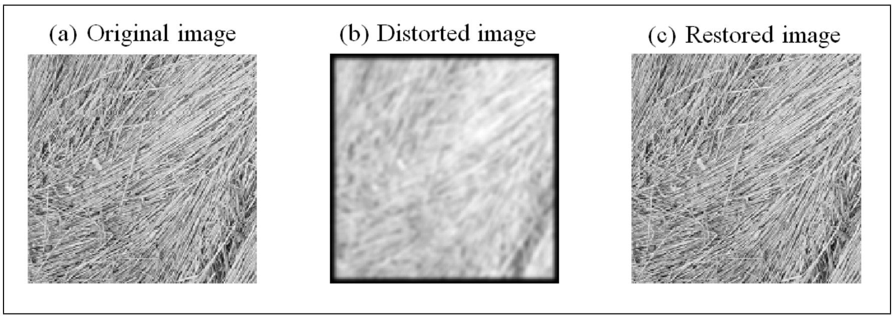


FIGURE 4.3: Blind Image Deconvolution using the developed method, for Example 4.1.

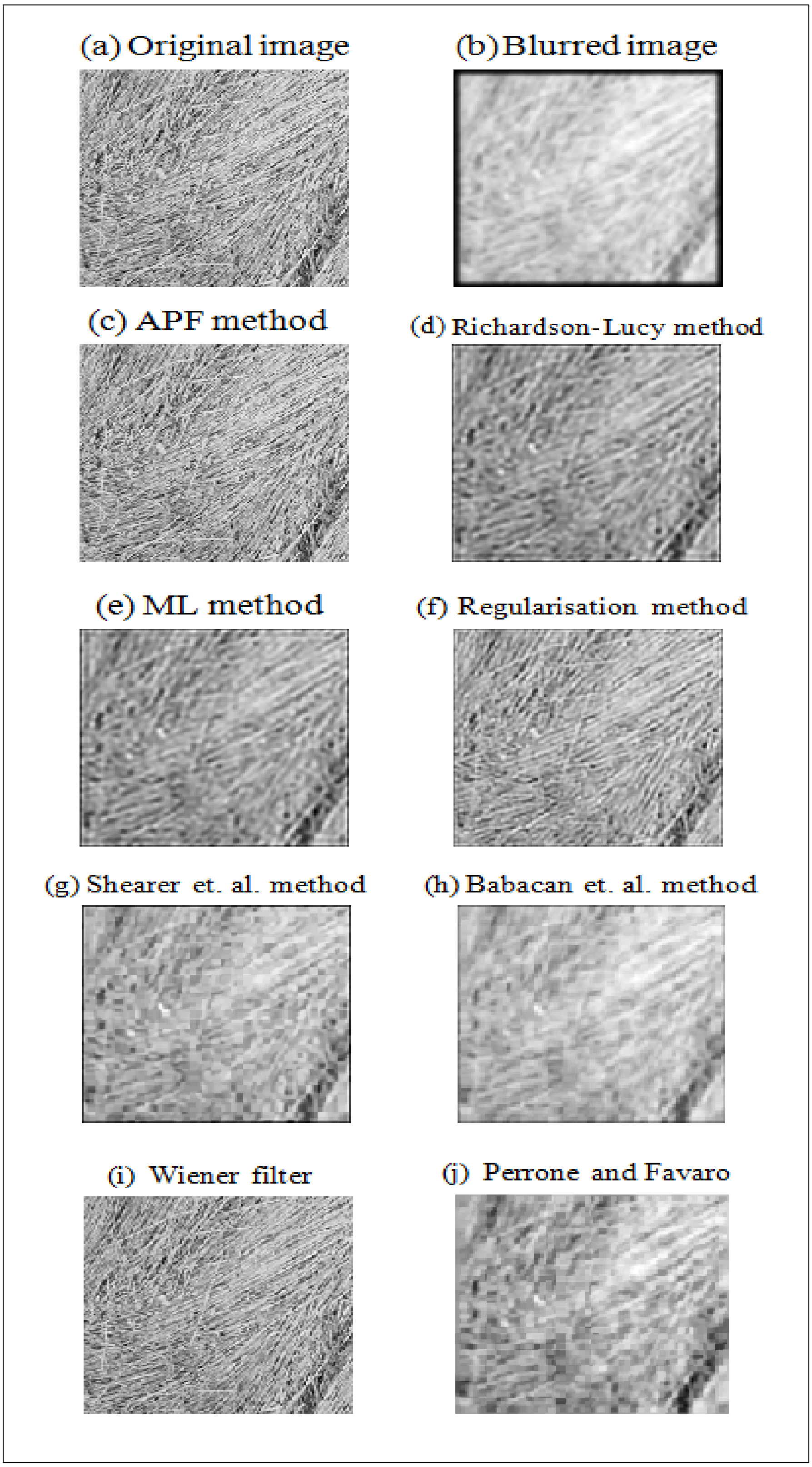

(a) Original image
(b) Blurred image
(c) APF method
(d) Richardson-Lucy method
(e) ML method
(f) Regularisation method
(g) Shearer et. al. method
(h) Babacan et. al. method
(i) Wiener filter
(j) Perrone and Favaro

FIGURE 4.4: Comparisons between the result of Example 4.1 using Algorithm 4.4 and existing methods. (a) An original image and (b) a blurred image obtained after the addition of measurement error to the PSF and additive noise. Deblurred images of the image in (b) obtained by (c) AGCD computations and APF. The deblurred images are obtained by (d) Richardson-Lucy algorithm [6], (e) maximum likelihood algorithm (ML) [7], (i) Wiener algorithm [8] and (f) the regularisation method [9]. These are implemented in the image processing toolbox in MATLAB. The deblurred images obtained through statistical methods are shown in (g) Shearer et al. [4], (h) Babacan et al. [5] and (j) Perrone and Favaro methods [3].

Table 4.2: Comparison table for Example 4.1.

BID Methods	Size of PSF	PSF	SNR	MSE	Relative error
Developed method (APF)	not specified	not specified	43.34	4.6×10^{-5}	8.5×10^{-3}
Richardson-Lucy	specified	specified	16.74	2.12×10^{-2}	1.95×10^{-1}
Wiener	specified	specified	24.33	3.69×10^{-3}	8.16×10^{-2}
ML	specified	specified	16.88	2.05×10^{-2}	1.94×10^{-1}
Regularisation	specified	specified	18.72	1.34×10^{-2}	1.65×10^{-1}
Shearer et al.	specified	not specified	16.26	2.37×10^{-2}	2.27×10^{-1}
Babacan et al.	specified	not specified	15.21	3.01×10^{-2}	2.27×10^{-1}
Perrone and Favaro	specified	not specified	16.94	2.02×10^{-2}	2.11×10^{-1}

Example 4.2. *Camera-man true image of size* 180×180 *pixels processed with a Gaussian blurring PSF of size* 25×25*.*

The distorted image produced using the above PSF and recovered image using Algorithm 4.4 are shown in Figure 4.7. Algorithm 4.4

The results of the developed AGCD method processing are given in Figures 4.5, 4.6 and 4.7. Figure 4.5 shows the distribution of the computed PSF row and column sizes. The most frequently computed rank (the mode) is 24 *for rows and columns, and therefore the size of the PSF is* 25×25*. Table 4.3 shows the mode, median and mean values for the distribution of the computed rank for both rows and columns. The table shows that the mode, median and mean are the same since all the computed rank values are the same.*

Figure 4.6 shows the computed PSF in comparison with the exact PSF, and it shows clearly that the developed method gives a good estimate of the PSF.

Figure 4.7 shows the restored image using the developed method. Figure 4.8 and Table 4.4 gives a comparison of the SNR, MSE and Relative error performance measures among the state-of-the-art BID methods. PSNR was not used since it did not show any significant differences among all the compared methods.

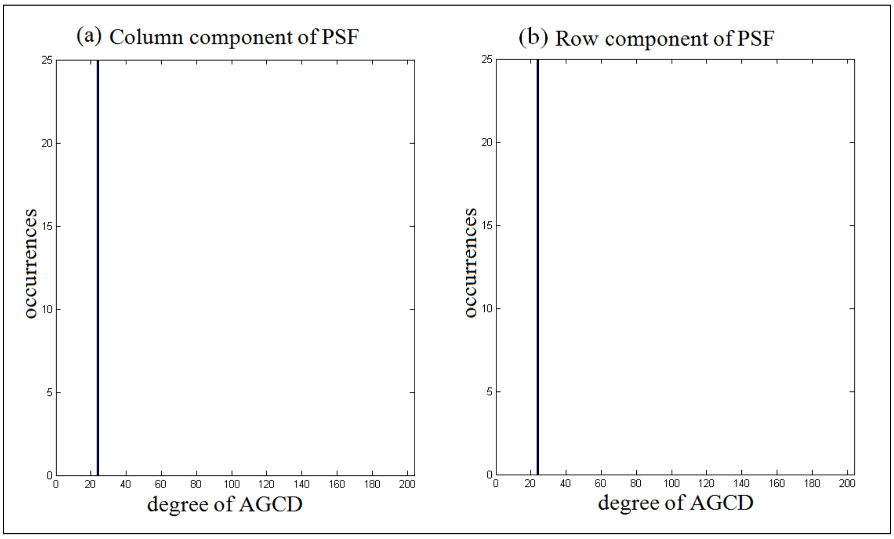


FIGURE 4.5: PSF's computed row and column size using the developed AGCD method for Example 4.2.

Table 4.3: PSF's computed degrees for Example 4.2.

Computing Rank	Mode	Median	Mean
Rows	24	24	24
Columns	24	24	24

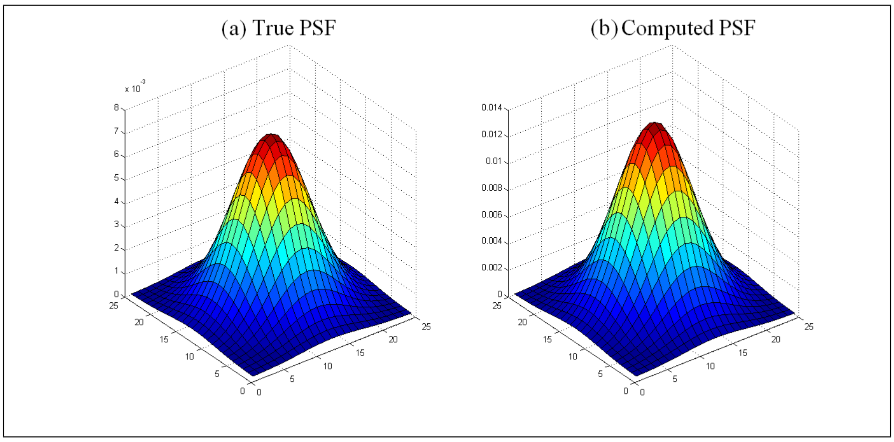


Figure 4.6: The true and computed PSFs that are applied to the distorted image in Example 4.2.

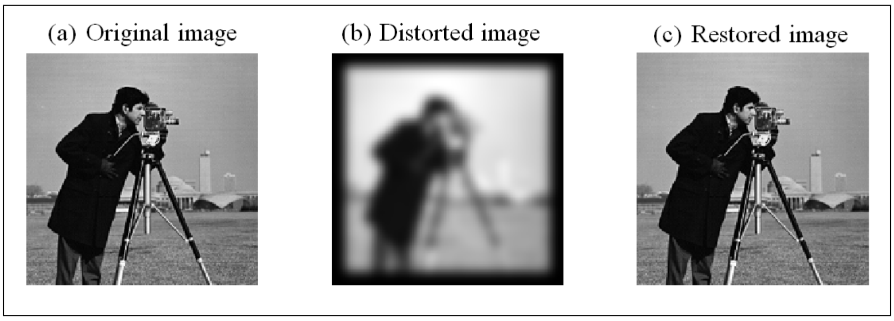


Figure 4.7: Blind Image Deconvolution using the developed method, for Example 4.2.

Table 4.4: Comparison table for Example 4.2.

BID Methods	Size of PSF	PSF	SNR	MSE	Relative error
Developed method (APF)	not specified	not specified	53.64	4×10^{-6}	2.20×10^{-3}
Richardson-Lucy	specified	specified	17.54	1.76×10^{-2}	2.25×10^{-1}
Wiener	specified	specified	22.34	5.84×10^{-3}	9.46×10^{-2}
ML	specified	specified	17.24	1.89×10^{-2}	2.20×10^{-1}
Regularisation	specified	specified	20.62	8.67×10^{-3}	1.68×10^{-1}
Shearer et al.	specified	not specified	15.90	2.57×10^{-2}	2.49×10^{-1}
Babacan et al.	specified	not specified	14.09	3.90×10^{-2}	2.51×10^{-1}
Perrone and Favaro	specified	not specified	13.68	4.29×10^{-2}	2.04×10^{-1}

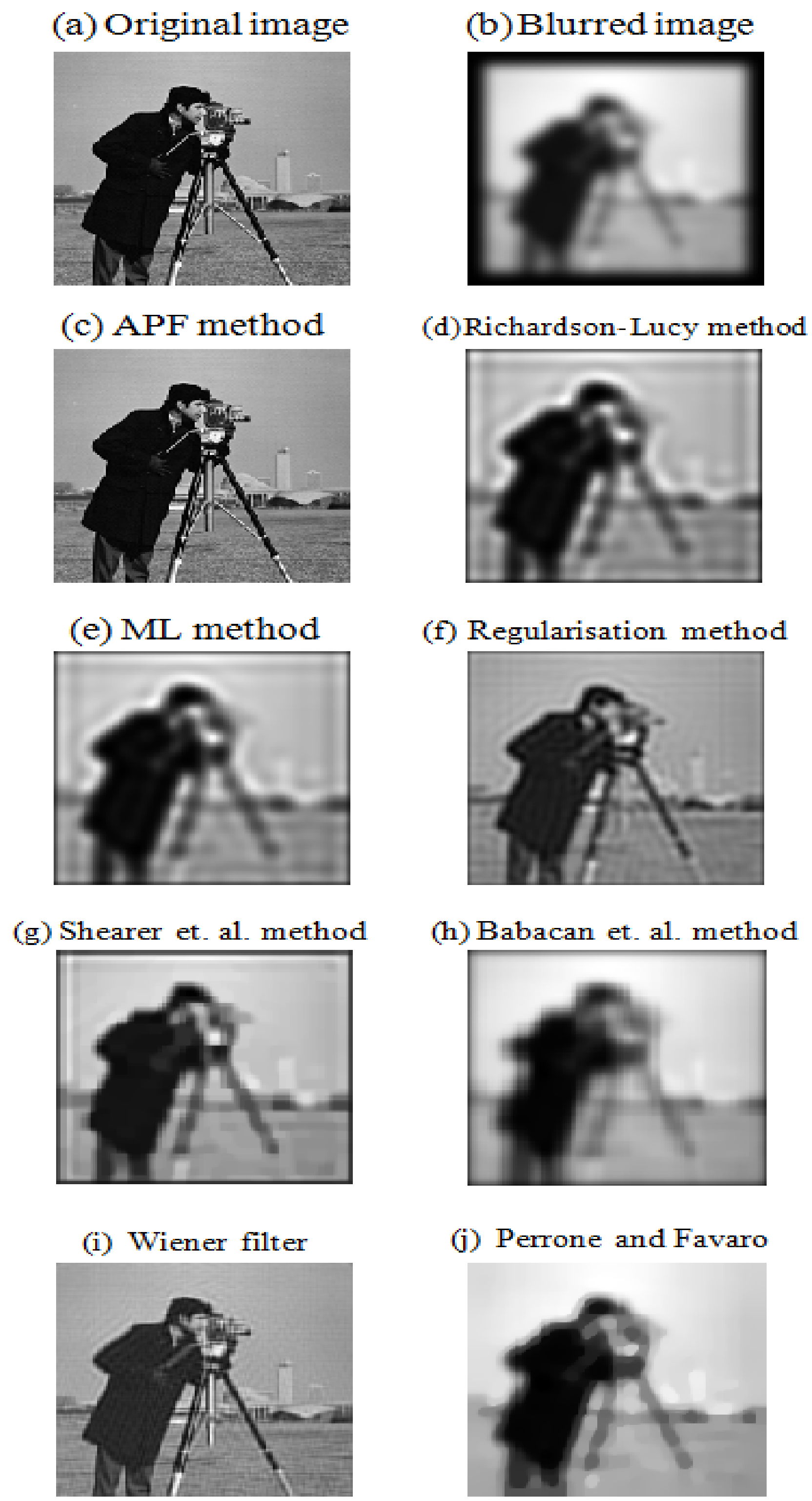

(a) Original image
(b) Blurred image
(c) APF method
(d)Richardson-Lucy method
(e) ML method
(f) Regularisation method
(g) Shearer et. al. method
(h) Babacan et. al. method
(i) Wiener filter
(j) Perrone and Favaro

FIGURE 4.8: Comparisons between the result of Example 4.2 using Algorithm 4.4 and existing methods. (a) An original image and (b) a blurred image obtained after the addition of measurement error to the PSF and additive noise. Deblurred images of the image in (b) obtained by (c) AGCD computations and APF. The deblurred images are obtained by (d) Richardson-Lucy algorithm [6], (e) maximum likelihood algorithm (ML) [7], (i) Wiener algorithm [8] and (f) the regularisation method [9].These are implemented in the image processing toolbox in MATLAB. The deblurred images obtained through statistical methods are shown in (g) Shearer et al. [4], (h) Babacan et al. [5] and (j) Perrone and Favaro methods [3].

Table 4.4 shows that the developed method has better estimation performance in terms of SNR, MSE and Relative error despite that the algorithm assumed no prior information of the PSF or its size.

Example 4.3. *Girl-face true image of size* 180 × 180 *pixels processed with a Gaussian blurring PSF of size* 35 × 35.

The distorted image produced using the above PSF and recovered image using Algorithm 4.4 are shown in Figure 4.11. Algorithm 4.4

The results of the developed AGCD method processing are given in Figures 4.9, 4.10 and 4.11. Figure 4.9 shows the distribution of the computed PSF row and column sizes. The most frequently computed rank (the mode) is 34 *for rows and columns, and therefore the size of the PSF is* 35 × 35. *Table 4.5 shows the mode, median and mean values for the distribution of the computed rank for both rows and columns. The table shows that the mode and median are the same but the mean is different, since all the computed rank values are not the same.*

Figure 4.10 shows the computed PSF in comparison with the exact PSF, and it shows clearly that the developed method gives a good estimate of the PSF.

Figure 4.11 shows the restored image using the developed method. Figure 4.12 and Table 4.6 gives a comparison of the SNR, MSE and Relative error performance measures among the state-of-the-art BID methods. PSNR was not used since it did not show any significant differences among all the compared methods.

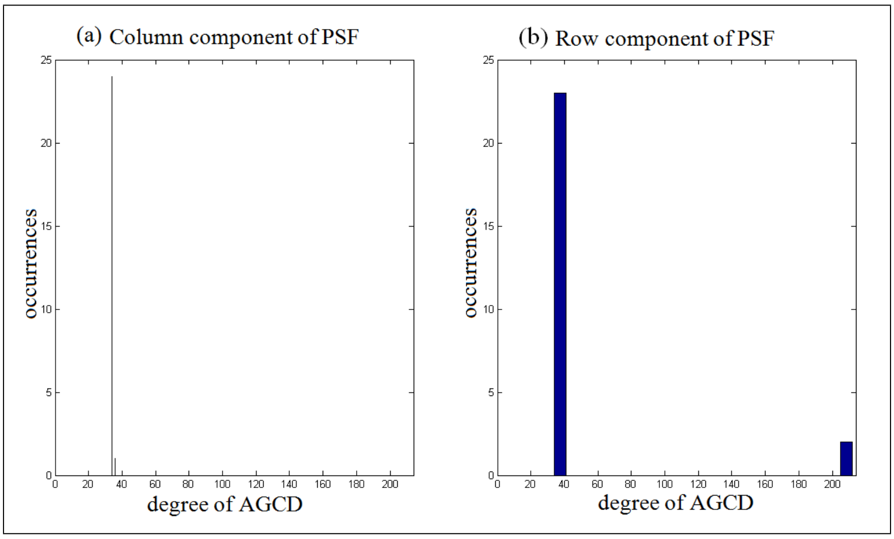


FIGURE 4.9: PSF's computed row and column size using the developed AGCD method for Example 4.3.

Table 4.5: PSF's computed degrees for Example 4.3.

Computing Rank	Mode	Median	Mean
Rows	34	34	48.24
Columns	34	34	34.080

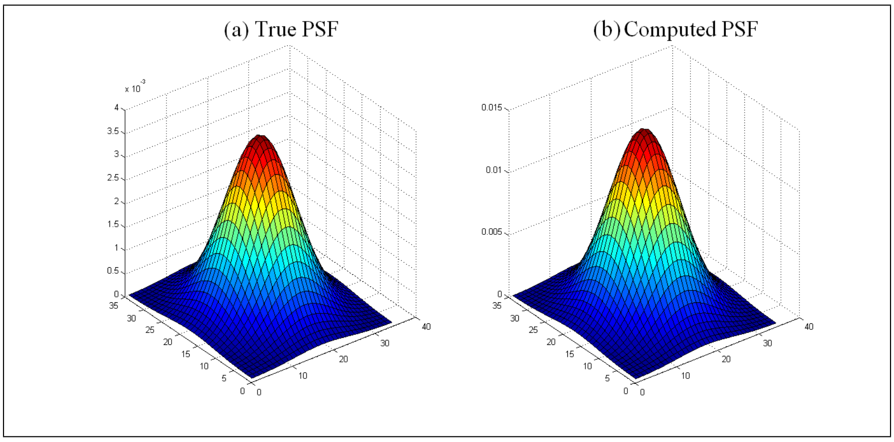


Figure 4.10: The true and computed PSFs that are applied to the distorted image in Example 4.3.

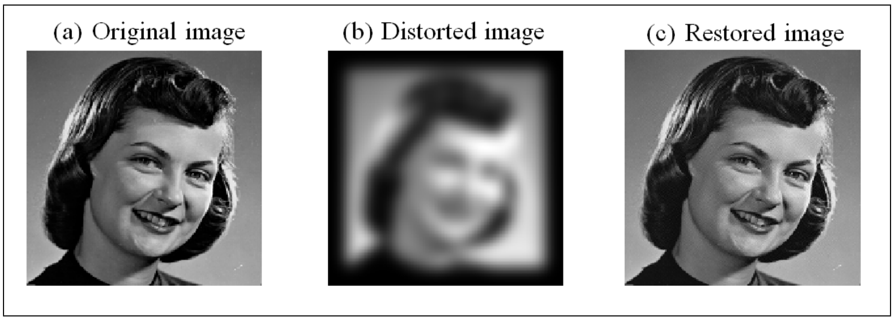


Figure 4.11: Blind Image Deconvolution using the developed method, for Example 4.3.

Table 4.6: Comparison table for Example 4.3.

BID Methods	Size of PSF	PSF	SNR	MSE	Relative error
Developed method (APF)	not specified	not specified	28.08	1.55×10^{-3}	2.80×10^{-2}
Richardson-Lucy	specified	specified	18.55	1.39×10^{-2}	2.64×10^{-1}
Wiener	specified	specified	21.78	6.63×10^{-3}	1.20×10^{-1}
ML	specified	specified	18.66	1.36×10^{-2}	2.55×10^{-1}
Regularisation	specified	specified	18.44	1.43×10^{-2}	1.78×10^{-1}
Shearer et al.	specified	not specified	15.73	2.67×10^{-2}	2.94×10^{-1}
Babacan et al.	specified	not specified	14.96	3.19×10^{-2}	3.07×10^{-1}
Perrone and Favaro	specified	not specified	14.92	3.22×10^{-2}	2.72×10^{-1}

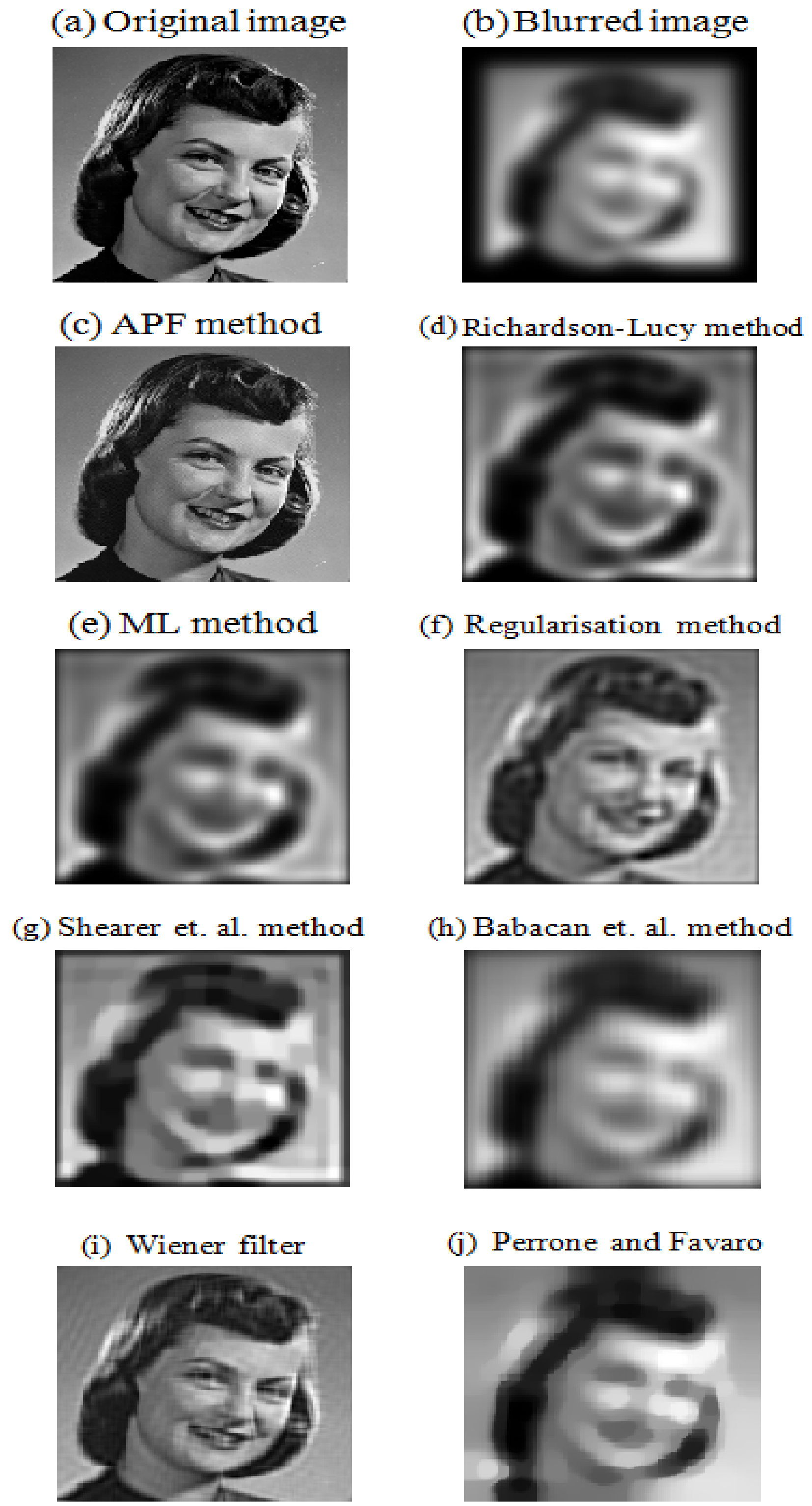

(a) Original image
(b) Blurred image
(c) APF method
(d) Richardson-Lucy method
(e) ML method
(f) Regularisation method
(g) Shearer et. al. method
(h) Babacan et. al. method
(i) Wiener filter
(j) Perrone and Favaro

FIGURE 4.12: Comparisons between the result of Example 4.3 using Algorithm 4.4 and existing methods. (a) An original image and (b) a blurred image obtained after the addition of measurement error to the PSF and additive noise. Deblurred images of the image in (b) obtained by (c) AGCD computations and APF. The deblurred images are obtained by (d) Richardson-Lucy algorithm [6], (e) maximum likelihood algorithm (ML) [7], (i) Wiener algorithm [8] and (f) the regularisation method [9]. These are implemented in the image processing toolbox in MATLAB. The deblurred images obtained through statistical methods are shown in (g) Shearer et al. [4], (h) Babacan et al. [5] and (j) Perrone and Favaro methods [3].

Table 4.6 shows that the developed method has better estimation performance in terms of SNR, MSE and Relative error despite that the algorithm assumed no prior information of the PSF or its size.

4.5 Summary

This chapter has considered the application of a solution of BID using linear algebra, specifically, an AGCD of two inexact polynomials. The method that has been discussed in this chapter considers the PSF as separable. The mathematical model of an image represents the image as a two-dimensional, in which its components are the pixel values of the image. The method discussed in this thesis works with a grayscale image that can be represented as a matrix. The matrix represents the image as bivariate polynomials and its coefficients are the components of the matrix. Therefore, the blurred image is the multiplication of two bivariate polynomials the original image and the PSF. Therefore, this consideration allows the separable PSF to be obtained by two separate AGCD computations. These operations are ill-posed and thus the AGCD computations must be done with care. Therefore, the AGCD computations are performed using an approximate polynomial factorisation by the Sylvester resultant matrix of two inexact polynomials that was discussed in Chapter 3. Then, by deconvolved the computed PSF from the blurred image, thereby obtaining an approximation of the original image.

The implementation of the BID theory for separable PSF is considered and some examples and results are included at the end of the chapter. the examples shows that the developed method has better estimation performance in terms of SNR, MSE and Relative error despite that the algorithm assumed no prior information of the PSF or its size.

The extension of the method to form a non-separable PSF from separable PSF is considered in the next chapter.

Chapter 5

Blind Image Deconvolution for non-Separable PSF

5.1 Introduction

Chapter 4 showed the developed BID method for a separable PSF and it was shown that a separable PSF can be calculated using one blurred image, and then a deblurred form of the blurred image can be obtained by deconvolving the computed separable PSF from the blurred image. However, the method discussed in Chapter 4 is not appropriate for non-separable PSF. This chapter shows that this problem requires two blurred images for non-separable PSF calculation. These two blurred images are required to have the same blurring PSF. These images may be obtained from one imaging system for two different scenes. This method is practical for situations where the PSF is caused by the camera.

The algorithm required for deblurring an image that is formed by a non-separable PSF is illustrated in the next section. The next chapter provides an extension of the developed work which reproduces the results by using the z-fourier transform and non-linear structure matrix method APF in a totally blind image deconvolution for a non-separable PSF.

5.2 The Extension of the Blind Image Deconvolution Method

This section demonstrates the computations that must be included for a non-separable PSF computation. The method that is presented in this section is considered to be a modified form of the developed method for the solution of the BID problem. Precisely, as was mentioned in Chapter 4, if the PSF is separable, the solution of the BID problem requires one blurred image, and the computation of the AGCD for the PSF computation requires the selection of two rows and two columns from the blurred image. In contrast, if the PSF is non-separable, the solution of the BID problem requires two blurred images, and the computation of the PSF requires the computation of the AGCD of the ith pair of rows and jth pair of columns of the blurred images.

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two different blurred images of $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively, and the blurred images are formed by the same spatially invariant and non-separable PSF $\mathcal{H}$. The bivariate polynomial form of $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{H}$ are G_1, G_2, F_1, F_2 and H, respectively. Then the convolution operation of the blurred images can be written as follows:

$$G_1(x, y) = H(x, y)F_1(x, y), \tag{5.1}$$

$$G_2(x, y) = H(x, y)F_2(x, y). \tag{5.2}$$

For simplicity, it is assumed that the G_1 and G_2 have the same size of $(M+p)\times(N+r)$, where the size of the exact images F_1 and F_2 are $M \times N$ and the size of the H is $(p+1)\times(r+1)$.

Consider $r_1(i)$ and $r_2(i)$ as the ith pair of rows of G_1 and G_2, respectively, and $c_1(j)$ and $c_2(j)$ are the jth pair of columns of G_1 and G_2

$$r_1(i) = G_1(i, y) = H(i, y)F_1(i, y), \tag{5.3}$$

$$r_2(i) = G_2(i, y) = H(i, y)F_2(i, y), \tag{5.4}$$

$$c_1(j) = G_1(x, j) = H(x, j)F_1(x, j), \tag{5.5}$$

$$c_2(j) = G_2(x, j) = H(x, j)F_2(x, j). \tag{5.6}$$

It is obvious that if the polynomial forms of $F_{1,2}(i, y)$ and $F_{1,2}(x, j)$ of $\mathcal{F}$ are coprime, then

$$H(i, y) = \text{AGCD}\Big(r_1(i), r_2(i)\Big) \qquad i = 1, \ldots, M + p, \tag{5.7}$$

$$H(x, j) = \text{AGCD}\Big(c_1(j), c_2(j)\Big) \qquad j = 1, \ldots, N + r. \tag{5.8}$$

Therefore, the AGCD algorithm of two univariate polynomials, which was discussed in Sections 3.3, 3.4 and 3.5, is applied to every pair of rows and every pair of columns of G_1 and G_2. There are then two sets of PSF $\mathcal{H}$. The first set is obtained by considering the rows of the blurred images G_1 and G_2, and the second set is obtained by considering the columns of G_1 and G_2. The components of the two sets of $\mathcal{H}$ are obtained by a sequence of independent computations of an AGCD along each row and column. Thus, every sequence of AGCDs has different scale factors associated with it. In order to compute the PSF $\mathcal{H}$, these scale factors must be removed. The method for their removal has been described in detail in [1, 18, 19].

Let $A(i, j)$ and $B(i, j)$ be matrices of order $(M + p) \times (N + r)$ of the coefficients of the AGCD computation that correspond to the two obtained sets of $\mathcal{H}$ using (5.7) and (5.8), respectively. Then a matrix of order $M \times N$ of the GCD computation.

$$A(i,j) = a'(i)H(i,j), \qquad i = 1,\ldots,M+p; j = 1,\ldots,N+r, \tag{5.9}$$

$$B(i,j) = b'(j)H(i,j), \qquad i = 1,\ldots,M+p; j = 1,\ldots,N+r, \tag{5.10}$$

where $a'(i)$ and $b'(j)$ are non-zero scaler multipliers, since every AGCD is defined with an arbitrary non-zero scaler multiplier. Therefore, $H(i,j)$ can be rewritten as:

$$A(i,j)a(i) = H(i,j), \qquad i = 1,\ldots,M+p; j = 1,\ldots,N+r, \tag{5.11}$$

$$B(i,j)b(j) = H(i,j), \qquad i = 1,\ldots,M+p; j = 1,\ldots,N+r, \tag{5.12}$$

where

$$a(i) = \frac{1}{a'(i)}, \tag{5.13}$$

$$b(i) = \frac{1}{b'(i)}. \tag{5.14}$$

Then, the elimination of $H(i,j)$ between (5.11) and (5.12) yields

$$A(i,j)a(i) - B(i,j)b(j) = 0, \qquad i = 1,\ldots,M+p; j = 1,\ldots,N+r, \tag{5.15}$$

In a matrix form, equation (5.15) can be written as

$$\begin{bmatrix} S_1 & -S_2 \end{bmatrix} y = 0, \qquad y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{5.16}$$

where $S_1 \in \mathbb{C}^{(M+p)(N+r)\times(M+p)}$ and $S_2 \in \mathbb{C}^{(M+p)(N+r)\times(N+r)}$; they are given by

$$S_1 = \begin{bmatrix} A(1,1) & 0 & 0 & \dots & 0 \\ A(1,2) & 0 & 0 & \dots & 0 \\ A(1,3) & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ A(1,N+r) & 0 & 0 & \dots & 0 \\ 0 & A(2,1) & 0 & \dots & 0 \\ 0 & A(2,2) & 0 & \dots & 0 \\ 0 & A(2,3) & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & A(2,N+r) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & A(M+p,1) \\ 0 & 0 & 0 & \dots & A(M+p,2) \\ 0 & 0 & 0 & \dots & A(M+p,3) \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & A(M+p,N+r) \end{bmatrix}, \tag{5.17}$$

and

$$S_2 = \begin{bmatrix} B(1,1) & 0 & 0 & \dots & 0 \\ 0 & B(1,2) & 0 & \dots & 0 \\ 0 & 0 & B(1,3) & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & B(1,N+r) \\ B(2,1) & 0 & 0 & \dots & 0 \\ 0 & B(2,2) & 0 & \dots & 0 \\ 0 & 0 & B(2,3) & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & B(2,N+r) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ B(M+p,1) & 0 & 0 & \dots & 0 \\ 0 & B(M+p,2) & 0 & \dots & 0 \\ 0 & 0 & B(M+p,3) & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & B(M+p,N+r) \end{bmatrix}, \quad (5.18)$$

respectively, and

$$y_1 = \begin{bmatrix} a(1) & a(2) & \dots & a(M+p) \end{bmatrix}^T \in \mathbb{C}^{M+p}, \quad (5.19)$$

$$y_2 = \begin{bmatrix} b(1) & b(2) & \dots & b(N+r) \end{bmatrix}^T \in \mathbb{C}^{N+r}. \quad (5.20)$$

Equation 5.15 has the coefficient matrix that has unit rank loss and the vector y that lies in its null space. Therefore, the solution of y can be computed by applying the singular value decompositions (SVD) of the coefficient of the matrix in (5.16). Hence, $H(i,j)$ is approximated from (5.11) and (5.12) by

$$H(i,j) = \frac{1}{2}\Big(A\,(i,j)\,a\,(i) + B\,(i,j)\,b\,(j)\Big), \quad (5.21)$$

where $i = 1, \ldots, M + p.$ and $j = 1, \ldots, N + r$. Furthermore, $H(i, j)$ is the computed $\mathcal{H}$ that is considered as the average of the two estimates of a blur $H(i, j)$. The first estimation of $H(i, j)$ is computed by considering the rows of the given degraded images, and the second estimation of $H(i, j)$ is computed by considering the columns of the given degraded images.

Finally, in order to recover the original images $\mathcal{F}_1$ and $\mathcal{F}_2$, the polynomial division is applied to deconvolve the component of the PSF $\mathcal{H}$ from $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively.

The algorithm and the results of the modified method are described in the next section.

5.3 Implementation

The implementation of the BID developed method for non-seprable PSF is considered in this section. The developed BID procedure for non-separable PSF starts with loading two raw blurred images that are formed by the same spatially invariant and non-separable PSF into Matlab using a standard function. To calculate the degree of the PSF, the blurred image is sampled for user selectable pairs of rows or columns. The number of pairs is selected at random. Each order of rows and columns is then used to calculate the degree of the PSF row-wise and column-wise between the two blurred images.

The degree used in the AGCD is then selected on the basis of the most common (mode) of the computed degrees. The computed degree from the pairs for spatially invariant PSF functions should be the same. However, noise present in the blurred images affects the computed values and hence the most common (mode) value is used as the degree of the PSF.

The coefficients of the non-separable PSF are then computed from all the selectable pairs of rows and columns of the two blurred images. The deblurred images are then deconvolved from the computed PSF. Algorithm 5.3 gives the Matlab implementation for Blind Image Deconvolution for non-separable and spatially invariant blur.

Algorithm 5.3: Developed Blind Image Deconvolution for Non-Separable and Spatially Invariant PSF

Input: Two blurred images $\mathcal{G}_1$ and $\mathcal{G}_2$.

Output: The deblurred images $\hat{\mathcal{F}}_1$ $\hat{\mathcal{F}}_1$ and computed PSF $\hat{\mathcal{H}}$.

Begin:

1. Calculate the size of $\mathcal{G}_1$ and $\mathcal{G}_2$, which is $M \times N$.

   % Calculate the size of the PSF.

   % Twenty five trials are used in the computation of the size of the PSF represented by the degree of AGCD

2. Define a random vector RR of size twenty five for rows random selection with values ranging from one to maximum size of $\mathcal{G}_1$ and $\mathcal{G}_2$ row.

3. Define a random vector RC of size twenty five for columns random selection with values ranging from one to maximum size of $\mathcal{G}_1$ and $\mathcal{G}_2$ column.

4. Initialise two vectors of size twenty five with zeros DR and DC to store the result of the calculated degrees of the AGCD.

5. **while** $i \leq 25$ **do**

6. $r1 = \mathcal{G}_1(RR(i), :);$

7. $r2 = \mathcal{G}_2(RR(i), :);$

8. Pre-process $r1$ and $r2$ using the algorithms detailed in section 3.4 which produces $\tilde{r1}_{\theta_0}$ and $\alpha_0 \tilde{r2}_{\theta_0}$.

```
9. Calculate the degree of AGCD($\tilde{r1}_{\theta_0}, \alpha_0\tilde{r2}_{\theta_0}$) using the QR decomposition of the Sylvester resultant matrix and its sub-resultant matrix $S(\tilde{r1}_{\theta_0}, \alpha_0\tilde{r2}_{\theta_0})$ as detailed in section 3.5.1.
10. Save the result in $DR(i)$.
11. end
12. while $i \leq 25$ do
13. $c1 = \mathcal{G}_1(:, RC(i))$;
14. $c2 = \mathcal{G}_2(:, RC(i))$;
15. Pre-process $c1$ and $c2$ using the algorithms detailed in section 3.4 which produces $\tilde{c1}_{\theta_0}$ and $\alpha_0\tilde{c2}_{\theta_0}$.
16. Calculate the degree of AGCD($\tilde{c1}_{\theta_0}, \alpha_0\tilde{c2}_{\theta_0}$) using the QR decomposition of the Sylvester resultant matrix and its sub-resultant matrix $S(\tilde{c1}_{\theta_0}, \alpha_0\tilde{c2}_{\theta_0})$ as detailed in section 3.5.1.
17. Save the result in $DC(i)$.
18. end
19. Set $Degree_R = mode(DR)$;
20. Set $Degree_C = mode(DC)$;
    % Calculate the entries of the PSF matrix represented by the coefficients of AGCD using $Degree_R$ and $Degree_C$.
    % Calculate the $M$ univariate AGCD for each row of $\mathcal{G}_1$ with the associated row of $\mathcal{G}_2$.
21. Define a matrix $A$ of size $M \times N$ to embed the computed AGCD for each row of $\mathcal{G}_1$ with the associated row of $\mathcal{G}_2$.
22. for $i = 1 : M$
    (a) $r1 = \mathcal{G}_1(i, :)$;
    (b) $r2 =$
        $mathcalG_2(i, :);$
    (c) Preprocess $r1$ and $r2$ resulting in $\tilde{r1}_{\theta_0}$ and $\alpha_0\tilde{r2}_{\theta_0}$ using the algorithms detailed in section 3.4.
    (d) Compute the coefficients of AGCD($\tilde{r1}_{\theta_0}, \alpha_0\tilde{r2}_{\theta_0}$) results in $d(i)$ as detailed in section 3.5.2.
    (e) Embed the $d(i)$ in the matrix $A$.
        $A(i, :) = d(i)$
    (f) end

    % Calculate the $N$ univariate AGCD for each column of $\mathcal{G}_1$ with the associated column of $\mathcal{G}_2$.

23. Define a matrix $B$ of size $M \times N$ to embed the computed AGCD for each row of $\mathcal{G}_1$ with the associated row of $\mathcal{G}_2$.
24. for $j = 1 : N$
    (a) $c1 = \mathcal{G}_1(:, j);$
    (b) $c2 = \mathcal{G}_2(:, j);$
    (c) Pre-process $c1$ and $c2$ using the algorithms detailed in section 3.4 which produces $\tilde{c1}_{\theta_0}$ and $\alpha_0\tilde{c2}_{\theta_0}$.
    (d) Compute the coefficients of AGCD($\tilde{c1}_{\theta_0}, \alpha_0\tilde{c2}_{\theta_0}$) results in $d(j)$ as detailed in section 3.5.2.
    (e) Embed the $d(j)$ in the matrix $B$.
        $B(:, j) = d(j)$
    (f) end

    % compute the PSF matrix as a product of $A$ and $B$.
25. Solve (5.16) using the singular value decomposition SVD of the coefficient matrix in (5.16) for calculating the solution of $a(i), i = 1, \ldots, M$ and $b(j), n = 1, \ldots, N$.
```

26. Compute the PSF $\hat{\mathcal{H}}$ using (5.21).

27. Compute the deblurred images $\hat{\mathcal{F}}_1$ and $\hat{\mathcal{F}}_1$ by using the division polynomial operation
$\hat{\mathcal{F}}_1 = \frac{\mathcal{G}_1}{\hat{\mathcal{H}}}$ and $\hat{\mathcal{F}}_2 = \frac{\mathcal{G}_2}{\hat{\mathcal{H}}}$.

End

Algorithm 5.3 is used in the following examples to deblur a pair of raw images produced from the same non-separable blur in their exact images. The blurred raw image is produced using (5.1) and (5.2).

The exact images are processed by non-separable Gaussian blurring PSF $\mathcal{H}$ with small error ($\mathcal{E}_1$ and $\mathcal{E}_2$) and then small noise is added to the result of the convolution of the process $\mathcal{N}_1$ and $\mathcal{N}_2$. However in this research study noise was not addressed and has been left for future investigations. In the following experiments very low and insignificant white noise levels have been added. These levels are impractical and cannot be used to justify the developed algorithm's performance with noise.

5.3.1 Example

This section contains three examples in which the distorted image is restored using Algorithm 5.3. Raw images of 120×120 pixels have been used to examine the developed method. The results of each of the examples are compared with other image deblurring methods described in Chapter 2 using performance measures as discussed in Section 2.5. The following examples show that the developed method resulted in a better image estimation performance compared with other methods described in Chapter 2.

Example 5.1. *Girl-face and lena true images of size* 180×180 *pixels processed with a non-separable Gaussian blurring PSF of size* 9×9.

The distorted images produced using the above PSF and recovered image using Algorithm 5.3 are shown in Figure 5.3. Algorithm 5.3

The results of the developed AGCD method processing are given in Figures 5.1, 5.2 and 5.3. Figure 5.1 shows the distribution of the computed PSF row and column sizes. The most frequently computed rank (the mode) is 8 *for rows and columns, and therefore the size of the PSF is* 9×9. *Table 5.1 shows the mode, median and mean values for the distribution of the computed rank for both rows and columns. The table shows that the mode, median and mean are the same since all the computed rank values are the same.*

Figure 5.2 shows the computed PSF in comparison with the exact PSF, and it shows clearly that the developed method gives a good estimate of the PSF.

Figure 5.3 shows the restored image using the developed method. Figures 5.4 and 5.5 and Tables 5.2 and 5.3 give a comparison of the SNR, MSE and Relative error performance measures among the state-of-the-art BID methods. PSNR was not used since it did not show any significant differences among all the compared methods.

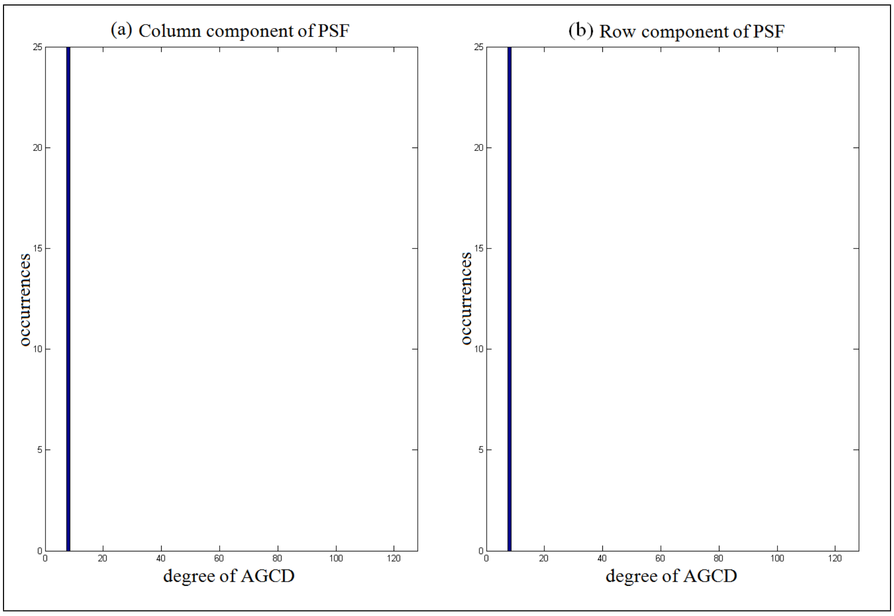


FIGURE 5.1: PSF's computed row and column size using the developed AGCD method for Example 5.1.

TABLE 5.1: PSF's computed degree for Example 5.1.

Computing Rank	Mode	Median	Mean
Rows	8	8	8
Columns	8	8	8

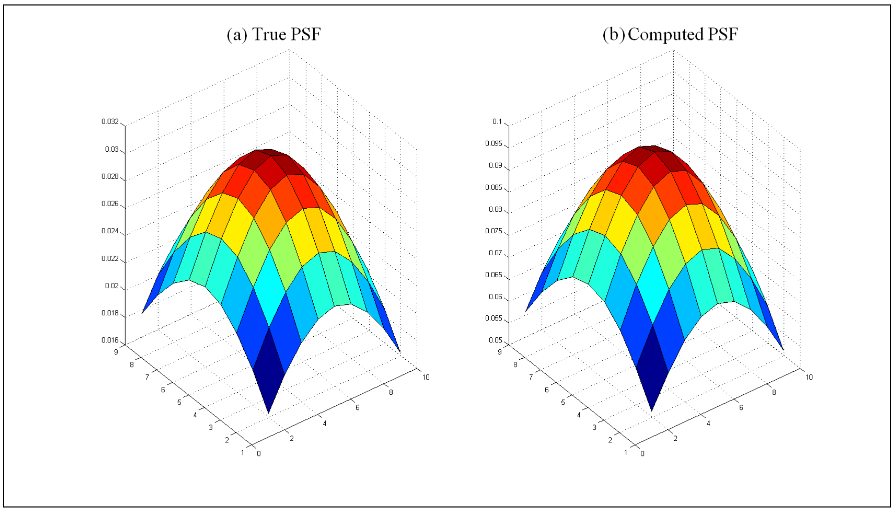


FIGURE 5.2: The true and computed PSFs that are applied to the distorted images in Example 5.1.

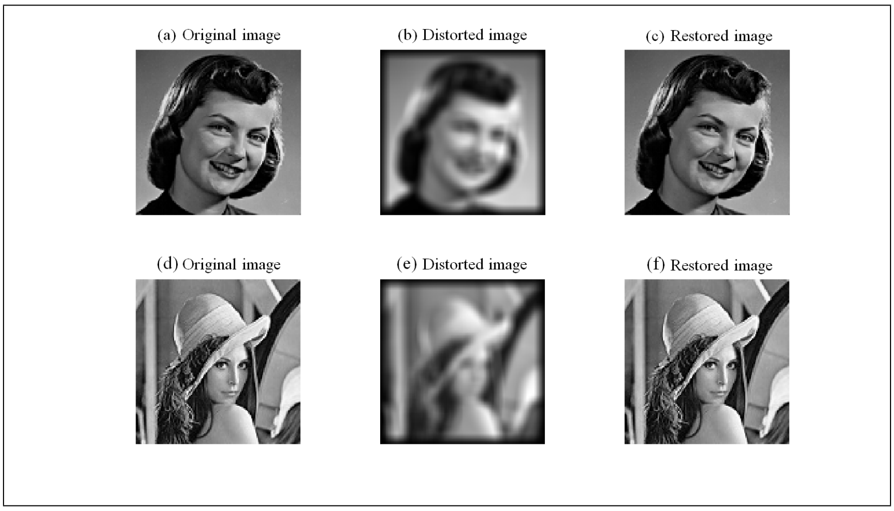


FIGURE 5.3: Blind Image Deconvolution using the developed method, for Example 5.1.

TABLE 5.2: Comparison table for the first distorted image in Example 5.1.

BID Methods	Size of PSF	PSF	SNR	MSE	Relative error
Developed method (APF)	not specified	not specified	77.17	1.92×10^{-8}	3.07×10^{-4}
Richardson-Lucy	specified	specified	20.37	9.17×10^{-3}	0.2115
Wiener	specified	specified	43.64	4.3×10^{-5}	6.34×10^{-3}
ML	specified	specified	12.80	0.0525	0.4037
Regularisation	specified	specified	22.11	6.16×10^{-3}	0.0862
Shearer et al.	specified	not specified	19.33	0.0117	0.2063
Babacan et al.	specified	not specified	18.67	0.0136	0.1918
Perrone and Favaro	specified	not specified	19.11	0.0123	0.1523

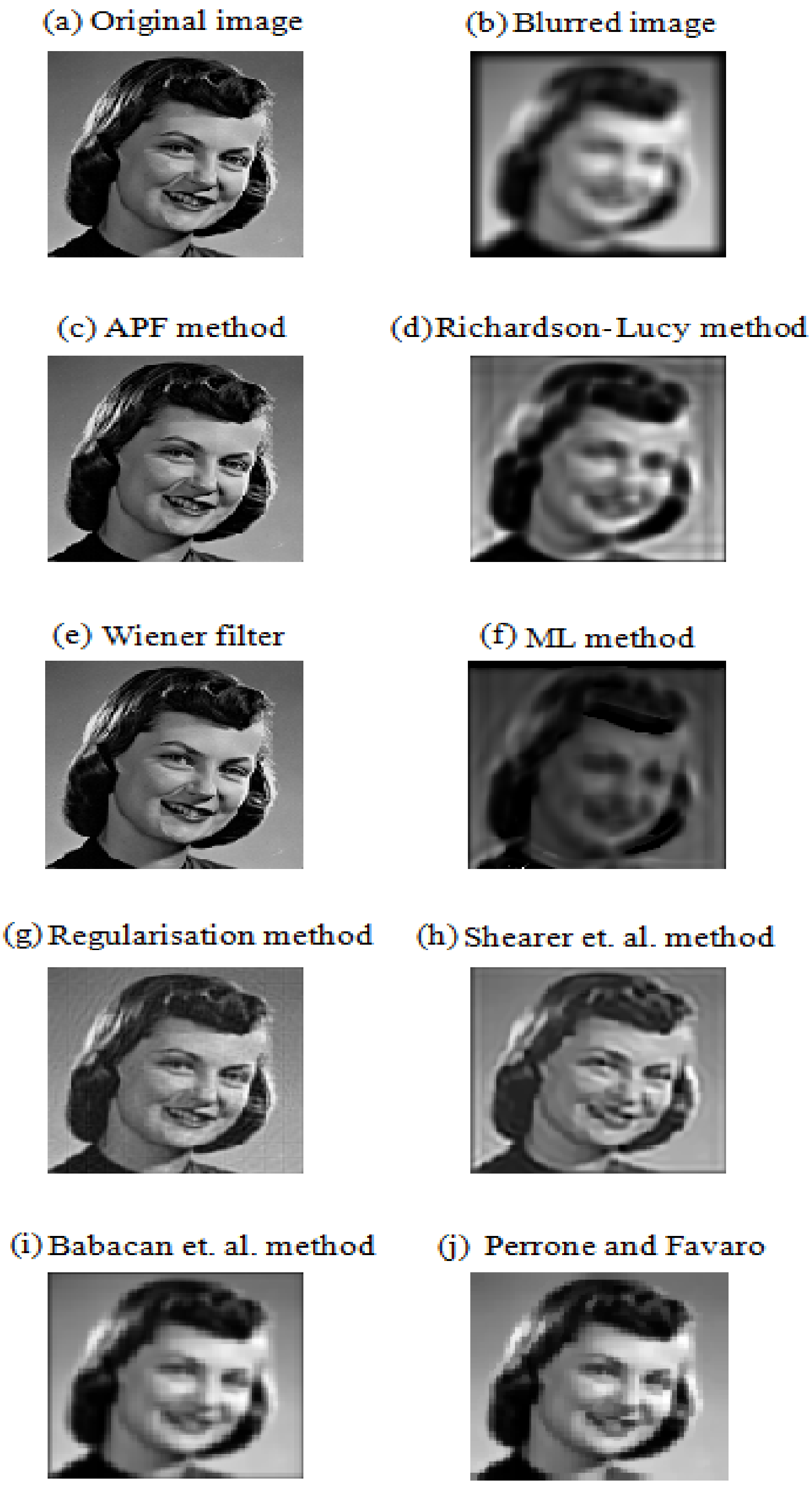

(a) Original image
(b) Blurred image
(c) APF method
(d)Richardson-Lucy method
(e) Wiener filter
(f) ML method
(g) Regularisation method
(h) Shearer et. al. method
(i) Babacan et. al. method
(j) Perrone and Favaro

FIGURE 5.4: Comparisons between the result of the first distorted image in Example 5.1 using Algorithm 5.3 and existing methods.(a) An original image and (b) a blurred image obtained after the addition of measurement error to the PSF and additive noise. Deblurred images of the image in (b) obtained by (c) AGCD computations and APF. The deblurred images are obtained by (d) Richardson-Lucy algorithm [6], (e) Wiener algorithm [8], (f) maximum likelihood algorithm (ML) [7] and (g) the regularisation method [9]. These are implemented in the image processing toolbox in MATLAB. The deblurred images obtained through statistical methods are shown in (h) Shearer et al. [4], (i) Babacan et al. [5] and (j) Perrone and Favaro methods [3].

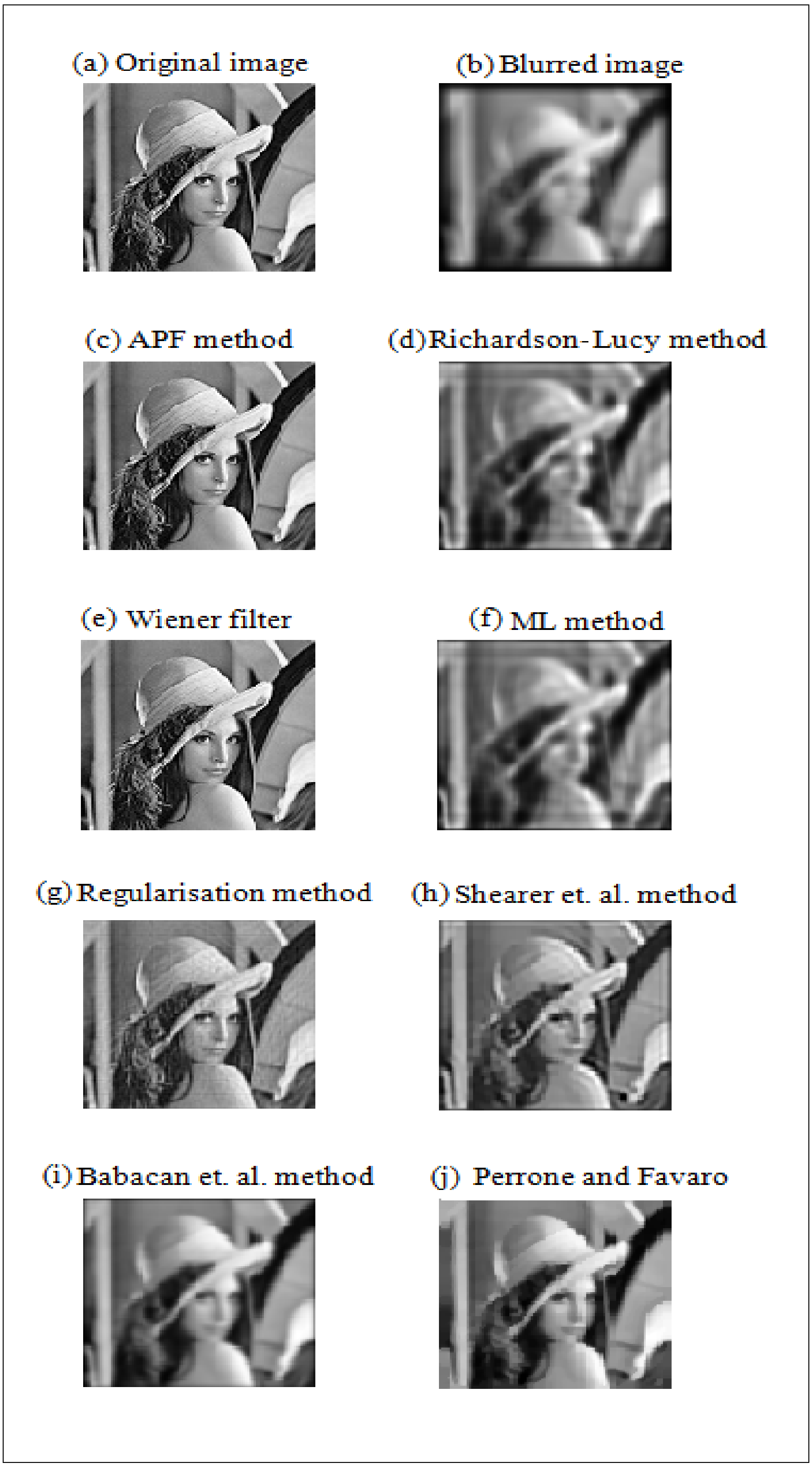

(a) Original image
(b) Blurred image
(c) APF method
(d)Richardson-Lucy method
(e) Wiener filter
(f) ML method
(g) Regularisation method
(h) Shearer et. al. method
(i) Babacan et. al. method
(j) Perrone and Favaro

FIGURE 5.5: Comparisons between the result of the second distorted image in Example 5.1 using Algorithm 5.3 and existing methods. (a) An original image and (b) a blurred image obtained after the addition of measurement error to the PSF and additive noise. Deblurred images of the image in (b) obtained by (c) AGCD computations and APF. The deblurred images are obtained by (d) Richardson-Lucy algorithm [6], (e) Wiener algorithm [8], (f) maximum likelihood algorithm (ML) [7] and (g) the regularisation method [9]. These are implemented in the image processing toolbox in MATLAB. The deblurred images obtained through statistical methods are shown in (h) Shearer et al. [4], (i) Babacan et al. [5] and (j) Perrone and Favaro methods [3].

Table 5.3: Comparison table for the second distorted image in Example 5.1.

BID Methods	Size of PSF	PSF	SNR	MSE	Relative error
Developed method (APF)	not specified	not specified	75.18	3.04×10^{-8}	2.58×10^{-4}
Richardson-Lucy	specified	specified	18.39	0.0144	0.2118
Wiener	specified	specified	46.71	2.1×10^{-5}	7.16×10^{-3}
ML	specified	specified	18.61	0.0137	0.2097
Regularisation	specified	specified	24.53	3.52×10^{-3}	0.0930
Shearer et al.	specified	not specified	20.27	9.4×10^{-3}	0.1754
Babacan et al.	specified	not specified	19.43	0.0114	0.1983
Perrone and Favaro	specified	not specified	21.17	7.6×10^{-3}	0.1621

Table 5.2 and 5.3 show that the developed method has better estimation performance in terms of SNR, MSE and Relative error despite that the algorithm assumed no prior information of the PSF or its size.

Example 5.2. *Girl and pumpkin true images of size* 180×180 *pixels processed with a non-separable Gaussian blurring PSF of size* 15×15.

The distorted images produced using the above PSF and recovered image using Algorithm 5.3 are shown in Figure 5.8. Algorithm 5.3

The results of the developed AGCD method processing are given in Figures 5.6, 5.7 and 5.8. Figure 5.6 shows the distribution of the computed PSF row and column sizes. The most frequently computed rank (the mode) is 14 *for rows and columns, and therefore the size of the PSF is* 15×15. *Table 5.4 shows the mode, median and mean values for the distribution of the computed rank for both rows and columns. The table shows that the mode and median are the same but the mean is different since all the computed rank values are not the same.*

Figure 5.7 shows the computed PSF in comparison with the exact PSF, and it shows clearly that the developed method gives a good estimate of the PSF.

Figure 5.8 shows the restored image using the developed method. Figures 5.9 and 5.10 and Tables 5.5 and 5.6 give a comparison of the SNR, MSE and Relative error performance measures among the state-of-the-art BID methods. PSNR was not used since it did not show any significant differences among all the compared methods.

Table 5.4: PSF's computed degree for Example 5.2.

Computing Rank	Mode	Median	Mean
Rows	14	14	14.04
Columns	14	14	14

Table 5.5 and 5.6 show that the developed method has better estimation performance in terms of SNR, MSE and Relative error despite that the algorithm assumed no prior information of the PSF or its size.

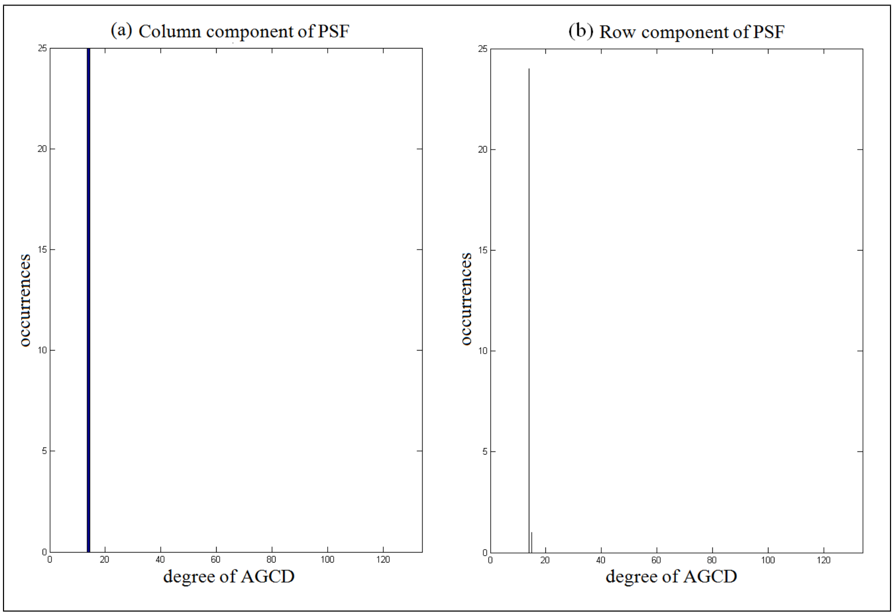


FIGURE 5.6: PSF's computed row and column size using the developed AGCD method for Example 5.2.

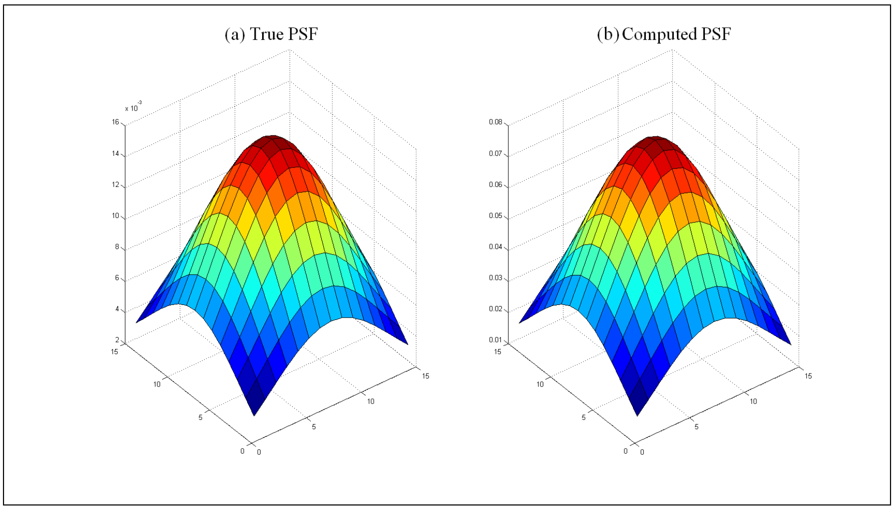


FIGURE 5.7: The true and computed PSFs that are applied to the distorted images in Example 5.2.

TABLE 5.5: Comparison table for the first distorted image in Example 5.2.

BID Methods	Size of PSF	PSF	SNR	MSE	Relative error
Developed method (APF)	not specified	not specified	36.11	2.45×10^{-4}	0.2464
Richardson-Lucy	specified	specified	17.45	0.0180	0.0342
Wiener	specified	specified	32.02	6.28×10^{-4}	0.2429
ML	specified	specified	17.61	0.0173	0.2848
Regularisation	specified	specified	22.47	5.66×10^{-3}	0.1317
Shearer et al.	specified	not specified	18.36	0.0146	0.2360
Babacan et al.	specified	not specified	17.50	0.0178	0.2590
Perrone and Favaro	specified	not specified	19.01	0.0126	0.2127

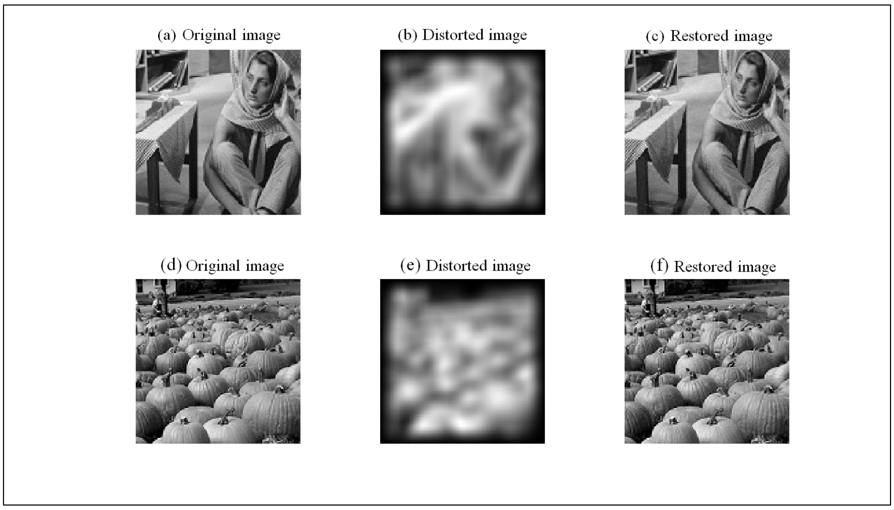


Figure 5.8: Blind Image Deconvolution using the developed method, for Example 5.2.

Table 5.6: Comparison table for the second distorted image in Example 5.2.

BID Methods	Size of PSF	PSF	SNR	MSE	Relative error
Developed method (APF)	not specified	not specified	37.19	1.91×10^{-4}	0.0276
Richardson-Lucy	specified	specified	16.84	0.0206	0.2870
Wiener	specified	specified	27.06	1.96×10^{-3}	0.0454
ML	specified	specified	16.79	0.0210	0.2848
Regularisation	specified	specified	18.66	0.0136	0.1778
Shearer et al.	specified	not specified	16.80	0.0209	0.2624
Babacan et al.	specified	not specified	15.66	0.0271	0.2813
Perrone and Favaro	specified	not specified	16.55	0.0221	0.2503

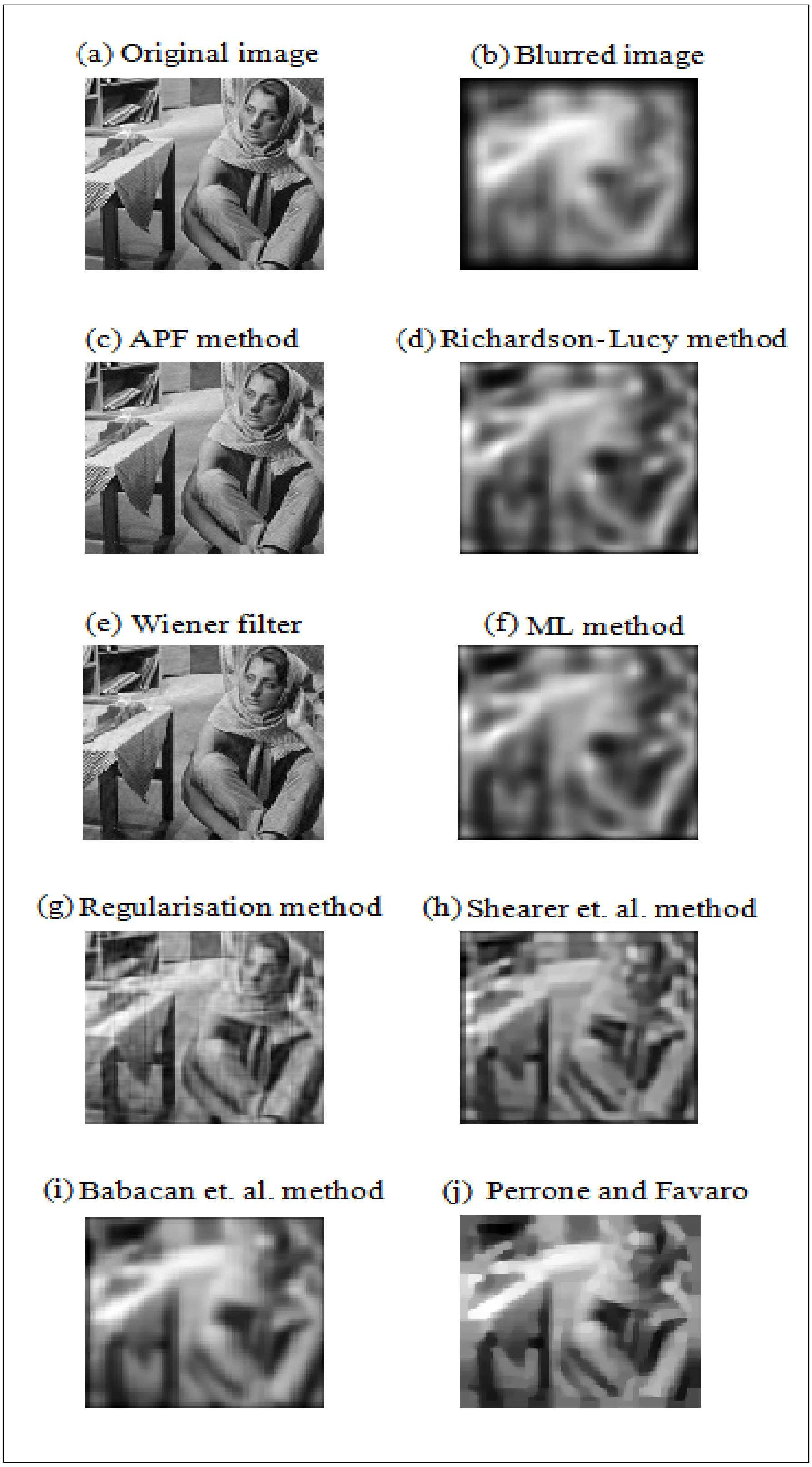

(a) Original image
(b) Blurred image
(c) APF method
(d) Richardson-Lucy method
(e) Wiener filter
(f) ML method
(g) Regularisation method
(h) Shearer et. al. method
(i) Babacan et. al. method
(j) Perrone and Favaro

FIGURE 5.9: Comparisons between the result of the first distorted image in Example 5.2 using Algorithm 5.3 and existing methods. (a) An original image and (b) a blurred image obtained after the addition of measurement error to the PSF and additive noise. Deblurred images of the image in (b) obtained by (c) AGCD computations and APF. The deblurred images are obtained by (d) Richardson-Lucy algorithm [6], (e) Wiener algorithm [8], (f) maximum likelihood algorithm (ML) [7] and (g) the regularisation method [9]. These are implemented in the image processing toolbox in MATLAB. The deblurred images obtained through statistical methods are shown in (h) Shearer et al. [4], (i) Babacan et al. [5] and (j) Perrone and Favaro methods [3].

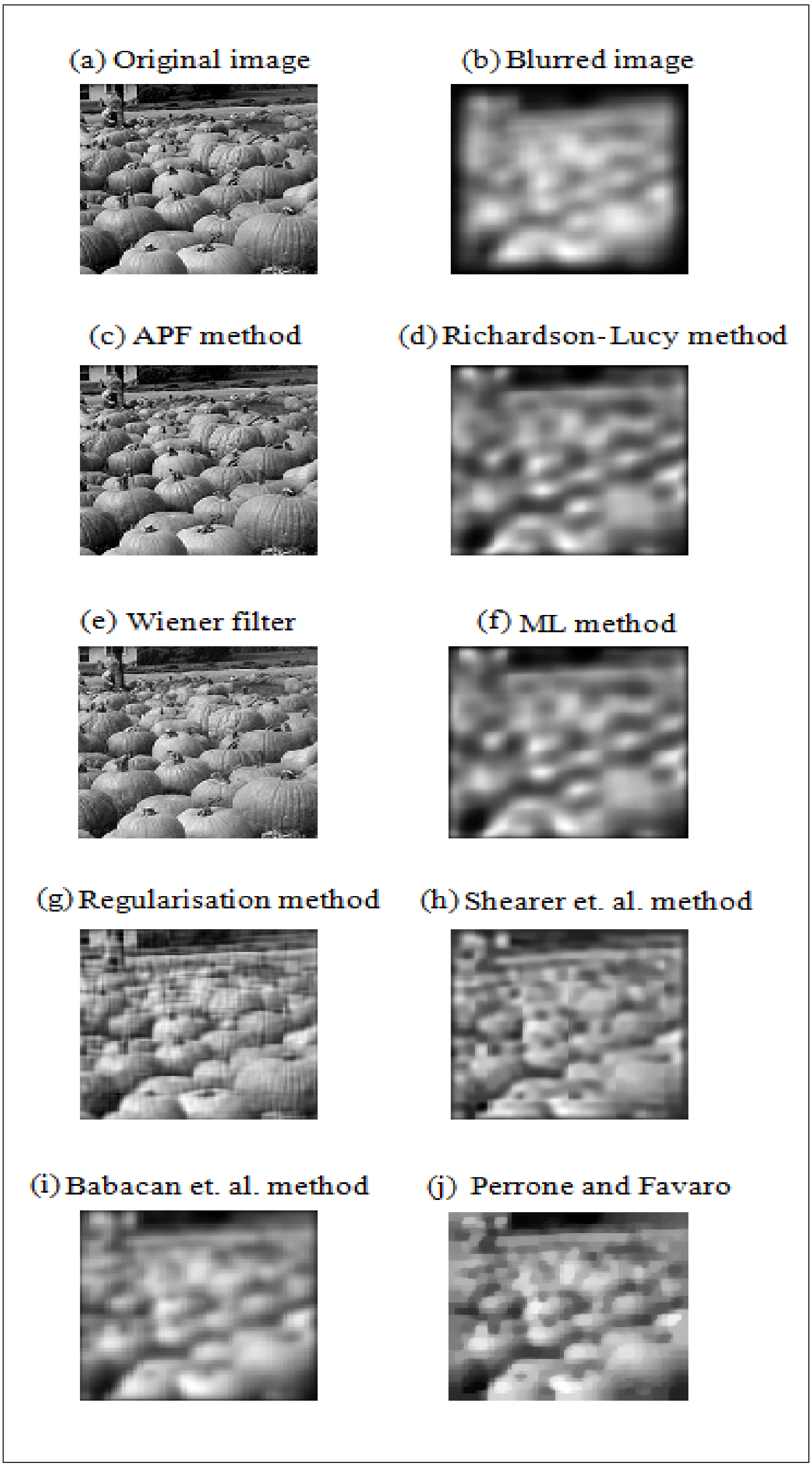

(a) Original image
(b) Blurred image
(c) APF method
(d) Richardson-Lucy method
(e) Wiener filter
(f) ML method
(g) Regularisation method
(h) Shearer et. al. method
(i) Babacan et. al. method
(j) Perrone and Favaro

FIGURE 5.10: Comparisons between the result of the second distorted image in Example 5.2 using Algorithm 5.3 and existing methods. (a) An original image and (b) a blurred image obtained after the addition of measurement error to the PSF and additive noise. Deblurred images of the image in (b) obtained by (c) AGCD computations and APF. The deblurred images are obtained by (d) Richardson-Lucy algorithm [6], (e) Wiener algorithm [8], (f) maximum likelihood algorithm (ML) [7] and (g) the regularisation method [9]. These are implemented in the image processing toolbox in MATLAB. The deblurred images obtained through statistical methods are shown in (h) Shearer et al. [4], (i) Babacan et al. [5] and (j) Perrone and Favaro methods [3].

Example 5.3. *Grass and camera-man true images of size* 180 × 180 *pixels processed with a non-separable Gaussian blurring PSF of size* 25 × 25*.*

The distorted images produced using the above PSF and recovered image using Algorithm 5.3 are shown in Figure 5.13. Algorithm 5.3

The results of the developed AGCD method processing are given in Figures 5.11, 5.12 and 5.13. Figure 5.11 shows the distribution of the computed PSF row and column sizes. The most frequently computed rank (the mode) is 24 *for rows and columns, and therefore the size of the PSF is* 25 × 25*. Table 5.7 shows the mode, median and mean values for the distribution of the computed rank for both rows and columns. The table shows that the mode, median and mean are the same since all the computed rank values are the same.*

Figure 5.12 shows the computed PSF in comparison with the exact PSF, and it shows clearly that the developed method gives a good estimate of the PSF.

Figure 5.13 shows the restored image using the developed method. Figures 5.14 and 5.15 and Tables 5.8 and 5.9 give a comparison of the SNR, MSE and Relative error performance measures among the state-of-the-art BID methods. PSNR was not used since it did not show any significant differences among all the compared methods.

Table 5.7: PSF's computed degree for Example 5.3.

Computing Rank	Mode	Median	Mean
Rows	24	24	24
Columns	24	24	24

Table 5.8 and 5.9 show that the developed method has better estimation performance in terms of SNR, MSE and Relative error despite that the algorithm assumed no prior information of the PSF or its size.

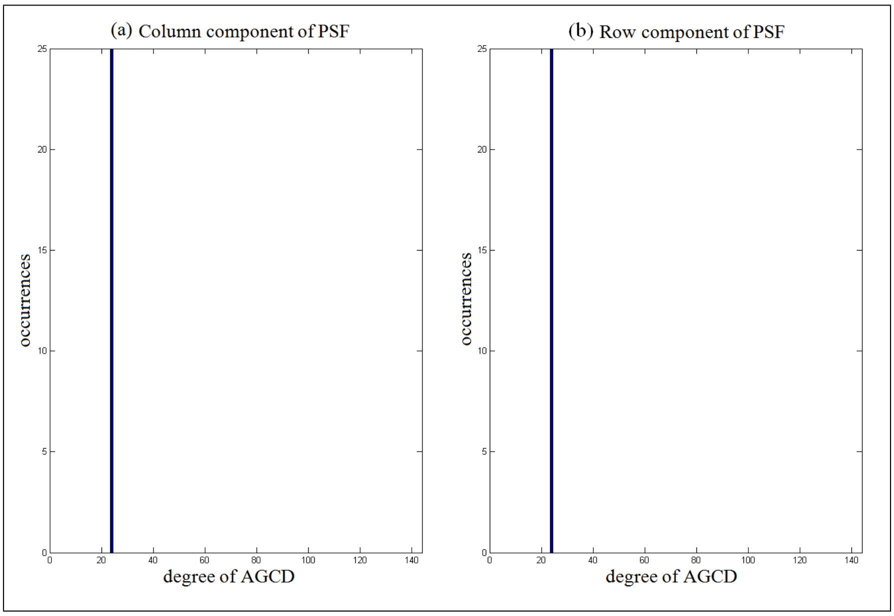


FIGURE 5.11: PSF's computed row and column size using the developed AGCD method for Example 5.3.

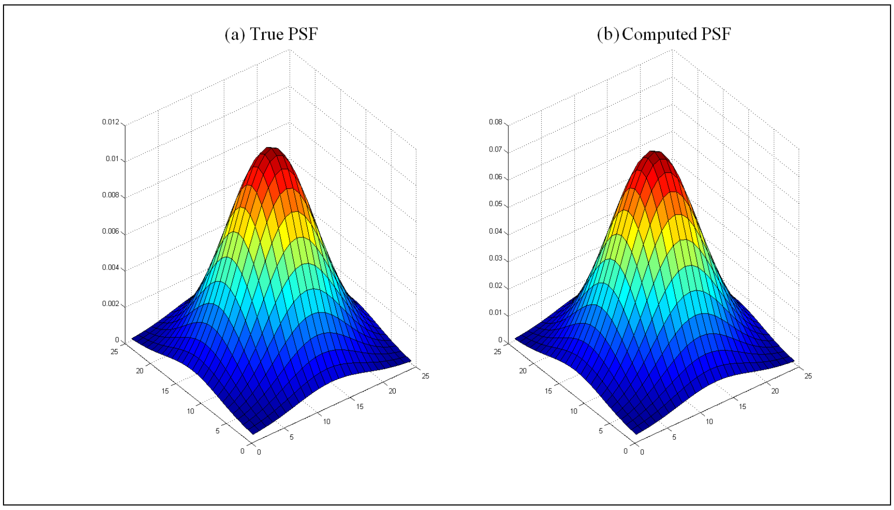


FIGURE 5.12: The true and computed PSFs that are applied to the distorted images in Example 5.3.

TABLE 5.8: Comparison table for the first distorted image in Example 5.3.

BID Methods	Size of PSF	PSF	SNR	MSE	Relative error
Developed method (APF)	not specified	not specified	43.01	5.0×10^{-5}	9.99×10^{-3}
Richardson-Lucy	specified	specified	15.93	0.0255	0.2823
Wiener	specified	specified	23.56	4.40×10^{-3}	0.0919
ML	specified	specified	15.63	0.0273	0.2714
Regularisation	specified	specified	16.86	0.0206	0.1829
Shearer et al.	specified	not specified	14.47	0.0357	0.3052
Babacan et al.	specified	not specified	13.71	0.0425	0.2863
Perrone and Favaro	specified	not specified	13.25	0.0474	0.2270

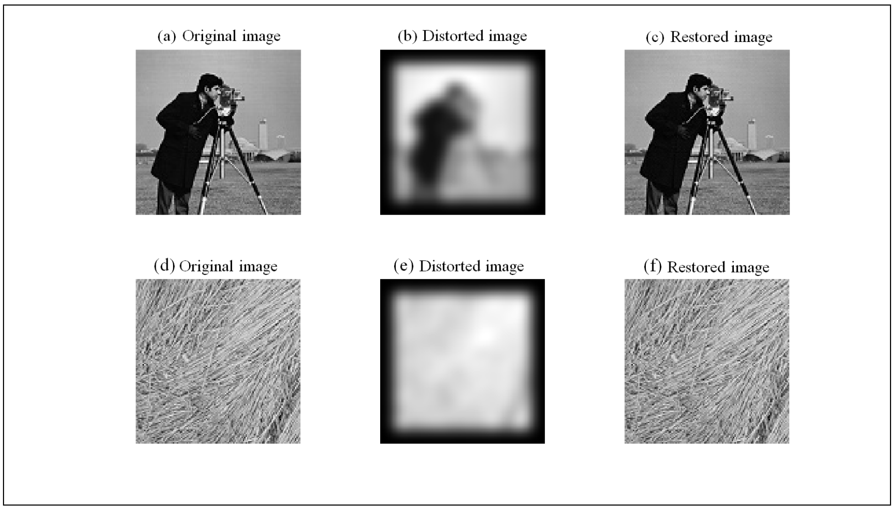


Figure 5.13: Blind Image Deconvolution using the developed method, for Example 5.3.

Table 5.9: Comparison table for the second distorted image in Example 5.3.

BID Methods	Size of PSF	PSF	SNR	MSE	Relative error
Developed method (APF)	not specified	not specified	35.78	2.64×10^{-4}	0.02478
Richardson-Lucy	specified	specified	15.46	0.0284	0.2399
Wiener	specified	specified	20.15	9.67×10^{-3}	0.1407
ML	specified	specified	15.12	0.0307	0.2330
Regularisation	specified	specified	16.26	0.0236	0.2003
Shearer et al.	specified	not specified	13.40	0.0457	0.2868
Babacan et al.	specified	not specified	12.44	0.0570	0.2750
Perrone and Favaro	specified	not specified	14.28	0.0373	0.3052

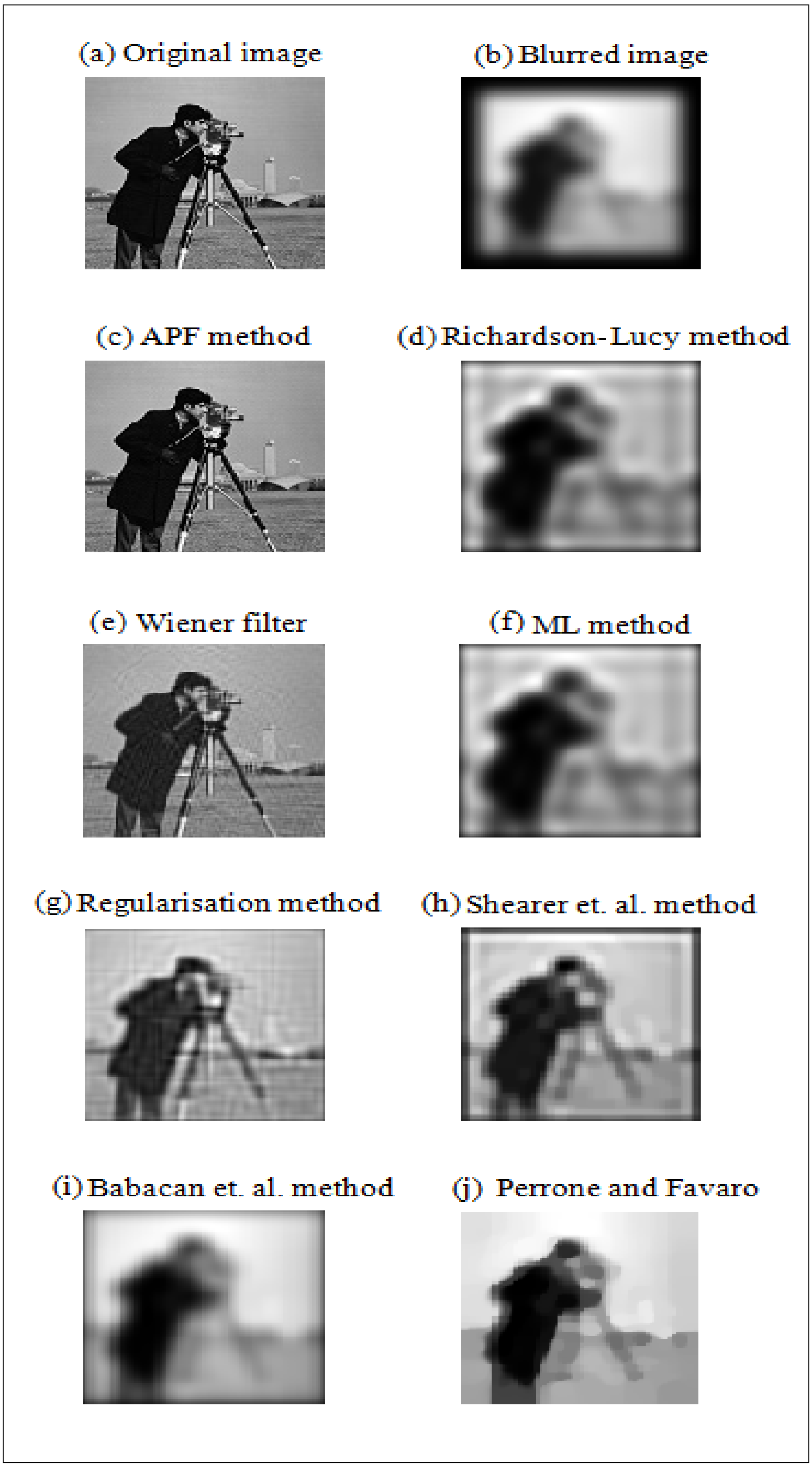

(a) Original image
(b) Blurred image
(c) APF method
(d) Richardson-Lucy method
(e) Wiener filter
(f) ML method
(g) Regularisation method
(h) Shearer et. al. method
(i) Babacan et. al. method
(j) Perrone and Favaro

FIGURE 5.14: Comparisons between the result of the first distorted image in Example 5.3 using Algorithm 5.3 and existing methods. (a) An original image and (b) a blurred image obtained after the addition of measurement error to the PSF and additive noise. Deblurred images of the image in (b) obtained by (c) AGCD computations and APF. The deblurred images are obtained by (d) Richardson-Lucy algorithm [6], (e) Wiener algorithm [8], (f) maximum likelihood algorithm (ML) [7] and (g) the regularisation method [9].These are implemented in the image processing toolbox in MATLAB. The deblurred images obtained through statistical methods are shown in (h) Shearer et al. [4], (i) Babacan et al. [5] and (j) Perrone and Favaro methods [3].

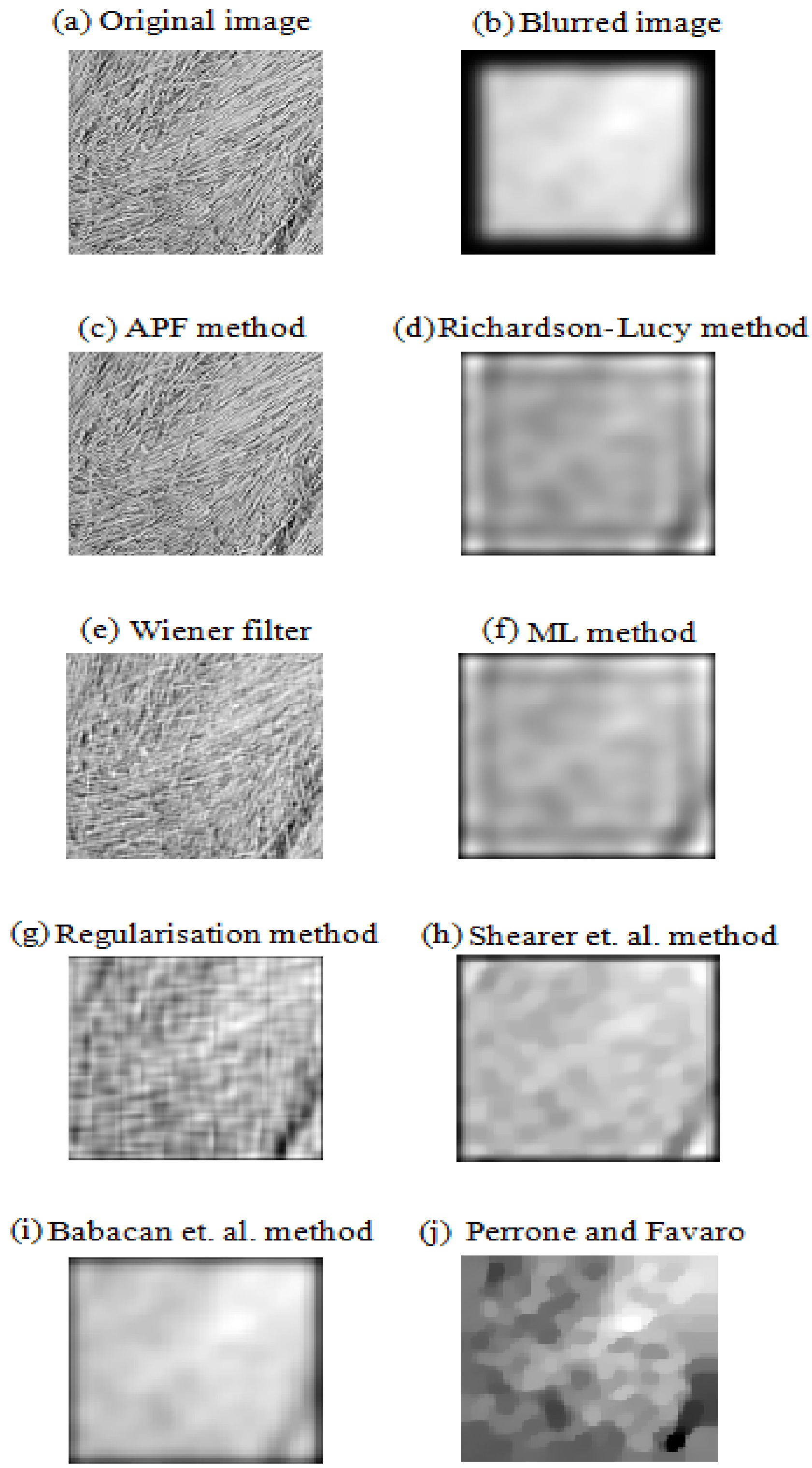

(a) Original image
(b) Blurred image
(c) APF method
(d) Richardson-Lucy method
(e) Wiener filter
(f) ML method
(g) Regularisation method
(h) Shearer et. al. method
(i) Babacan et. al. method
(j) Perrone and Favaro

FIGURE 5.15: Comparisons between the result of the second distorted image in Example 5.3 using Algorithm 5.3 and existing methods. (a) An original image and (b) a blurred image obtained after the addition of measurement error to the PSF and additive noise. Deblurred images of the image in (b) obtained by (c) AGCD computations and APF. The deblurred images are obtained by (d) Richardson-Lucy algorithm [6], (e) Wiener algorithm [8], (f) maximum likelihood algorithm (ML) [7] and (g) the regularisation method [9]. These are implemented in the image processing toolbox in MATLAB. The deblurred images obtained through statistical methods are shown in (h) Shearer et al. [4], (i) Babacan et al. [5] and (j) Perrone and Favaro methods [3].

5.4 Summary

This chapter has considered the application of a BID solution to non-separable blur using linear algebra, specifically, the AGCD of two inexact polynomials. The method discussed in this chapter is appropriate for spatially invariant non-separable blur PSFs. In contrast to Chapter 4 showed the developed BID method for a separable PSF and it was shown that a separable PSF can be calculated using one blurred image, and then a deblurred form of the blurred image can be obtained by deconvolving the computed separable PSF from the blurred image. However, using one image is not appropriate for non-separable PSF and it requires two blurred images for the PSF calculation.

The implementation of this algorithm for non-separable PSF is discussed and some examples and results are included at the end of the chapter. The examples considered in this chapter examined blurs that are nearly separable which means they are simple cases of non-separable blur. The complex non-separable blur using motion blur combined with Gaussian blur can not be solve with this BID method. This research show that for complex non-separable blur frequency domain implementation of the method is more appropriate. Applying the z-fourier transform and non-linear structure matrix method APF in a blind image deconvolution is more appropriate for a non-separable PSF. The extension of the developed work that reproduced the results by using the frequency domain and APF in a totally blind image deconvolution for a complex non-separable PSF is provided in the next chapter.

Chapter 6

Blind Image Deconvolution for Non-Separable PSF using the Z-Fourier Transform

6.1 Introduction

Chapter 5 showed the developed BID method for a non-separable PSF, and it was shown that it can be calculated from the AGCD of two different blurred images. The cases considered in Chapter 5 used non-complex non-separable blurring functions. Blurring functions that have complex non-separable structure required the spectral domain implementation of the method developed in Chapter 5. In this research study It was found that the frequency domain implementation of the method is more suitable and applicable for non-separable PSF estimation, in general, and complex structure non-separable function, in particular. The spatial domain implementation of the method as presented in Chapter 5 did not perform well with non-separable PSF. The implementation of the method in the frequency domain solved the non-separable case of the PSF with better estimation results.

As has been mentioned in Chapter 5, this research requires two different blurred images that have been blurred by the same PSF to produce the blurred images. In real examples, this scenario works in cases where the blurring function occurs in the impulse response of the camera system (see Figure 6.1). However, the related work in [18, 19, 38] uses a different scenario in order to recover the true image. They assume that the true image can be calculated as the AGCD of two blurred images, in which two blurred images of the same scene are given. In practice, it is difficult to provide two blurred images that are perfectly aligned (see Figure 6.2) and in this case the theory of linear algebra is not appropriate to solve the problem of BID. In other words, it is important that every row and column that correspond to each other in the two blurred images have the same coordinates. Then, the row and column components in the true image can be computed as the AGCD between each row and column that correspond to each other in the two blurred images. To compare the performance of the BID methods based on the theory of linear algebra [1, 2] with the developed method, the developed method was modified to recover the true image by providing another blurred version of the same scene. The modified method was improved to reproduce the result by using z-Fourier transform and a non-linear structure matrix method APF in a totally blind image deconvolution for a

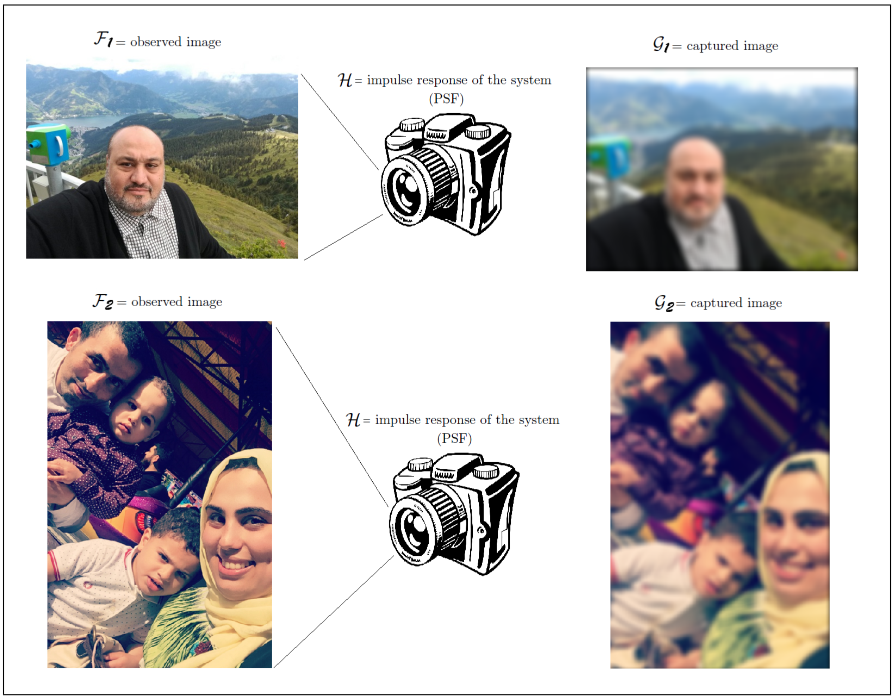

$\mathcal{F}_1$ = observed image
$\mathcal{H}$ = impulse response of the system (PSF)
$\mathcal{G}_1$ = captured image
$\mathcal{F}_2$ = observed image
$\mathcal{H}$ = impulse response of the system (PSF)
$\mathcal{G}_2$ = captured image

FIGURE 6.1: Real-life model of two blurred images captured using the same camera.

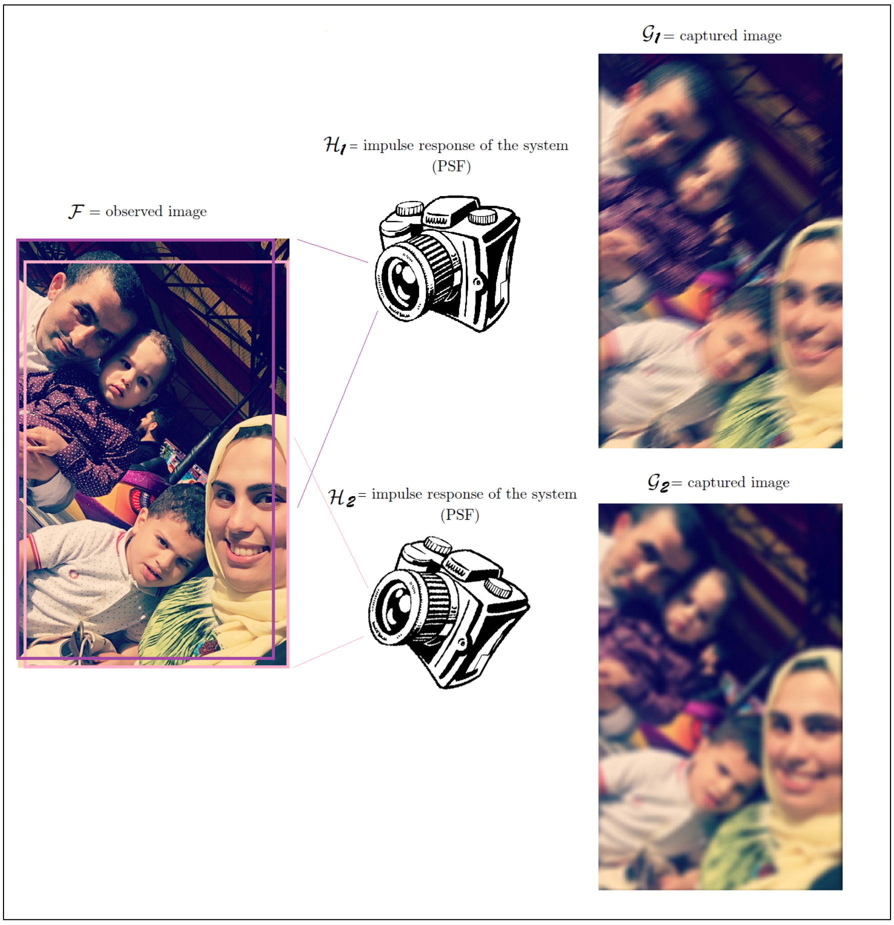

$\mathcal{G}_1$ = captured image
$\mathcal{H}_1$ = impulse response of the system (PSF)
$\mathcal{F}$ = observed image
$\mathcal{G}_2$ = captured image
$\mathcal{H}_2$ = impulse response of the system (PSF)

FIGURE 6.2: Real-life model of two blurred images of the same scene using different cameras.

non-separable PSF. This chapter will start by presenting a brief introduction about the z-Fourier transform. The modified method and its performance will then be illustrated.

6.2 Introduction of Z-Fourier Transform

The Fourier transform is one of the most commonly used techniques in control theory and signal processing. It is a one-to-one transform since it transforms the signal from a time-domain $f(x)$ to a frequency-domain $\mathbb{F}(n)$. Its benefit is that the z-Fourier transform allows the analysis of a signal through its frequency content.

The $1D$ Fourier transform $\mathbb{F}(n)$ of the function values $f(x)$ at the discrete points $x = 0, 1, \ldots, N-1$ can be defined as

$$\mathbb{F}(n) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp(-2\pi i x \frac{n}{N}), \qquad n = 0, 1, \ldots, N-1, \tag{6.1}$$

where $\mathbb{F}(n)$ is a periodic function of period N because the counter n ranges from 0 to $N-1$ and $\mathbb{F}(n+N) = \mathbb{F}(n)$. Additionally, it is observed that the z-Fourier transform value of $\mathbb{F}(n)$ corresponds to the value of the polynomial $\hat{f}(z)$. It is defined as

$$\hat{f}(z) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) z^x, \tag{6.2}$$

where z is a complex number

$$z = \exp(-2\pi i \frac{n}{N}), \qquad n = 0, 1, \ldots, N-1. \tag{6.3}$$

Then, the z-transform of the function values $f(x)$ is the function $N\hat{f}(z)$ for $x = 0, 1, \ldots, N-1$.

In the image processing, the $2D$ Fourier transform is commonly used. It can be defined as the $2D$ Fourier transform of a $M \times N$. The image is

$$\mathbb{F}(m, n) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp(-2\pi i x \frac{m}{M}) \exp(-2\pi i y \frac{n}{N}) \tag{6.4}$$

$$= \frac{1}{M} \sum_{x=0}^{M-1} \left[\ \tfrac{1}{N} \textstyle\sum_{y=0}^{N-1} f(x, y) \exp(-2\pi i y \tfrac{n}{N})\ \right] \exp(-2\pi i x \frac{m}{M}), \tag{6.5}$$

where $m = 0, 1, \ldots, M-1$ and $n = 0, 1, \ldots, N-1$. And because of

$$\mathbb{F}(m + M, n) = \mathbb{F}(m, n + N) = \mathbb{F}(m + M, n + N) = \mathbb{F}(m, n), \tag{6.6}$$

it means that the $\mathbb{F}(m, n)$ is a periodic function of the period M and N and the $2D$ Fourier transform is formed by two $1D$ Fourier transforms. Equation (6.5) shows that the function values of the z-Fourier transform $\mathbb{F}(m, n)$ correspond to the values of the bivariate polynomial $\hat{f}(z_1, z_2)$,

$$\hat{f}(z_1, z_2) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) z_1^x z_2^y, \tag{6.7}$$

where z_1 and z_2 are complex numbers

$$z_1 = \exp(-2\pi i \frac{m}{M}) \qquad \text{and} \qquad z_2 = \exp(-2\pi i \frac{n}{N}), \tag{6.8}$$

where $m = 0, 1, \ldots, M-1$ and $n = 0, 1, \ldots, N-1$. The $2D$ z-Fourier transform of the function values $f(x, y)$ of $x = 0, 1, \ldots, M-1$ and $y = 0, 1, \ldots, N-1$ is the function $MNf(z_1, z_2)$.

This research uses a Vandermonde matrix to compute the z-Fourier transform. The Vandermonde matrix is one class of the structured matrices and one of its applications in the signal processing field is the computation of the z-Fourier transform and Inverse z-Fourier Transform. The Vandermonde matrix can be applied to the z-Fourier transform as in the following:

Let's define the functions of $1D$ z-Fourier transform in (6.2) in a matrix form at the N^{th} roots of unity as

$$\hat{f}(z)_N = \begin{bmatrix} \hat{f}(0) \\ \hat{f}(1) \\ \hat{f}(2) \\ \vdots \\ \hat{f}(N-1) \end{bmatrix}, \tag{6.9}$$

$$f(x)_N = \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \\ f(N-1) \end{bmatrix}, \tag{6.10}$$

and the Vandermonde matrix V_N

$$V_N = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & z_N & z_N^2 & \dots & z_N^{N-1} \\ 1 & z_N^2 & z_N^4 & \dots & z_N^{2(N-1)} \\ 1 & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_N^{N-1} & z_N^{2(N-1)} & \dots & z_N^{(N-1)(N-1)} \end{bmatrix}. \tag{6.11}$$

These matrices allow the z-Fourier transform to be expressed as follows

$$\hat{f}(z)_N = V_N f(x)_N. \tag{6.12}$$

The V_N matrix is called the z-Fourier transform matrix. This research applied a Vandermonde matrix to compute the $2D$ z-Fourier transform. The next section illustrates the developed BID by using z-Fourier transform and APF for a non-separable PSF.

6.3 The Modification of the Blind Image Deconvolution Mmethod using DFT

This section demonstrates the computations that must be included for a non-separable PSF computation using z-Fourier transform. The method that is presented in this section is considered to be a modified form of the developed method that was discussed in Chapter 5. This modification aims to examine the feasibility of the Sylvester matrix and APF for blind image deconvolution.

This section is concerned with solving the BID problem when the extra knowledge required for the restoration of a true image is provided by another blurred version of the same scene; this means that it requires two blurred images of the same scene. The true image will be the AGCD of the two blurred images. The computation of AGCD in this case requires the use of the Fourier Transform to be applied.

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two blurred versions of the true $\mathcal{F}$, and the blurred images be formed by spatially invariant and non-separable PSF $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively.

The bivariate polynomial form of $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{F}$, $\mathcal{H}_1$ and $\mathcal{H}_2$ are G_1, G_2, F, H_1 and H_2, respectively. The convolution operation of the blurred images can then be written as follows:

$$G_1(x,y) = H_1(x,y)F(x,y), \tag{6.13}$$

$$G_2(x,y) = H_2(x,y)F(x,y). \tag{6.14}$$

For simplicity, it is assumed that the G_1 and G_2 have the same size of $(M+p)\times(N+r)$, where the size of the exact images F are $M \times N$ and the size of the H is $(p+1)\times(r+1)$.

Then the z-Fourier transform of the blurred images is

$$\mathbb{G}_1(z_1,z_2) = \sum_{x=1}^{M+p}\sum_{y=1}^{N+r} b_1(x,y)z_1^x z_2^y, \tag{6.15}$$

$$\mathbb{G}_2(z_1,z_2) = \sum_{x=1}^{M+p}\sum_{y=1}^{N+r} b_2(x,y)z_1^x z_2^y. \tag{6.16}$$

The convolution operation of the blurred images in the $2D$ z-transform can then be written as follows:

$$\mathbb{G}_1(z_1,z_2) = \mathbb{H}_1(z_1,z_2)\mathbb{F}(z_1,z_2), \tag{6.17}$$

$$\mathbb{G}_2(z_1, z_2) = \mathbb{H}_2(z_1, z_2)\mathbb{F}(z_1, z_2). \tag{6.18}$$

The aim of this application is to determine the true image, thus, it is assumed that the additive noise and measurement error are equal to zero. The substitution z_1 in (6.17) and (6.18) by its value in (6.8) then yields $M + p$ univariate polynomials $\mathbb{G}_1$ and $\mathbb{G}_1$ in the variable z_2,

$$\mathbb{G}_1(e^{-2\pi i \frac{m}{M+p}}, z_2) = \mathbb{H}_1(e^{-2\pi i \frac{m}{M+p}}, z_2)\mathbb{F}(e^{-2\pi i \frac{m}{M+p}}, z_2), \tag{6.19}$$

$$\mathbb{G}_2(e^{-2\pi i \frac{m}{M+p}}, z_2) = \mathbb{H}_2(e^{-2\pi i \frac{m}{M+p}}, z_2)\mathbb{F}(e^{-2\pi i \frac{m}{M+p}}, z_2), \tag{6.20}$$

for all $m = 1, 2, \ldots, M + p$. It follows that the univariate polynomial $\mathbb{F}$ is the GCD of $\mathbb{G}_1$ and $\mathbb{G}_2$ for each value of m.

$$\text{AGCD}\bigg(\mathbb{G}_1(m, z_2), \mathbb{G}_2(m, z_2))\bigg) = \mathbb{F}(m, z_2)\text{AGCD}\bigg(\mathbb{H}_1(m, z_2), \mathbb{H}_2(m, z_2))\bigg) \qquad m = 1, \ldots, M + p,$$

As long as, with high probability, $\mathbb{H}_1(m, z_2)$ and $\mathbb{H}_2(m, z_2)$ are coprime, thus

$$\text{AGCD}\bigg(\mathbb{H}_1(m, z_2), \mathbb{H}_2(m, z_2))\bigg) = c(m) \qquad m = 1, \ldots, M + p, \tag{6.22}$$

where $c(m)$ is a constant for each value of m. The substitution of (6.22) into (6.21) allows the $M + p$ univariate polynomial $\mathbb{F}(m, z_2)$ to be calculated, provided that the degree and the coefficient of AGCDs of $\mathbb{G}_1(m, z_2)$ and $\mathbb{G}_2(m, z_2)$ can be calculated using the QR decomposition and APF methods, respectively. The computation of the AGCD has been discussed in Chapter 4.

$$\text{AGCD}\bigg(\mathbb{G}_1(m, z_2), \mathbb{G}_2(m, z_2)\bigg) = \mathbb{F}(m, z_2)c(m) \qquad m = 1, \ldots, M + p, \tag{6.23}$$

in which $\mathbb{F}(m, z_2)$ is the mth row of the z-transform of the true image that is associated with the AGCD of the mth row of the the z-transform of the blurred image. Then, for each value m, the value of z_2 in (6.23) is substituted by its value in (6.8) from $n = 1, 2, \dots, N + r$.

$$A(m, n) = \mathbb{F}(e^{-2\pi i \frac{m}{M+p}}, e^{-2\pi i \frac{n}{N+r}})c(m). \tag{6.24}$$

It follows that the column version of the exact image allows it to be similarly computed by substituting z_2 in $\mathbb{G}_1$ and $\mathbb{G}_2$. Then, the $N + r$ univariate polynomials $\mathbb{F}(z_1, n)$ can be calculated as their AGCD. After that, by substituting z_1 by its value in (6.8), it will obtain another matrix

$$B(m, n) = \mathbb{F}(e^{-2\pi i \frac{m}{M+p}}, e^{-2\pi i \frac{n}{N+r}})k(n), \tag{6.25}$$

where $B(m, n)$ is the z-Fourier transform of the true image that is associated with the AGCD of the nth column of the z-transform of the blurred image. Whereas, $k(n)$ is a constant that is a function of n. Equation (6.24) and (6.25) yields

$$A(m, n)a(m) - B(m, n)b(n) = 0, \qquad m = 1, \dots, M + p; n = 1, \dots, N + r, \tag{6.26}$$

where

$$a(m) = \frac{1}{c(m)}, \tag{6.27}$$

$$b(n) = \frac{1}{k(n)}. \tag{6.28}$$

It clearly shows that (6.26) is identical to (5.15). Thus, the functions $a(m)$ and $b(n)$ are computed similarly by applying SVD to (5.16).

Hence, the estimated Fourier transform of the true image is then computed by

$$\mathbb{F} = \frac{1}{2}\Bigg(A\left(m,n\right)a\left(m\right) + B\left(m,n\right)b\left(n\right)\Bigg), \tag{6.29}$$

Finally, the $2D$ Inverse z-Fourier Transform of (6.29) yields an estimate of the true image $\mathcal{F}$. The algorithm and the result of the modified method, which computed the true image from two blurred versions of an image using AGCD, is described in the next section.

6.4 Implementation

The implementation of the BID algorithm for non-separable PSF using the structure matrix and z-Fourier transform is considered in this section. The developed BID procedure for non-separable PSF starts by loading a raw blurred image formed by the two non-separable PSFs into Matlab using a standard function.

To calculate the true image, z-Fourier transform is applied to the two blurred images using a Vandermonde matrix. Then, the degree of the true image is computed by selecting pairs of rows or columns from the blurred images. The number of pairs is selected at random. Each order of rows and columns is then used to calculate the degree of the true image row-wise and column-wise between the two blurred images.

The degree used in the AGCD is then selected on the basis of the most common (mode) of the computed degrees. The coefficients of the true image in the z-Fourier transform are then computed from all pairs of rows and columns of the two blurred images. The deblurred images are then deconvolved by applying the Inverse z-Fourier Transform. Algorithm 6.4 gives the Matlab implementation for the developed BID using the structure matrix and z-Fourier transform.

Algorithm 6.4: Developed Blind Image Deconvolution using the structure matrix and z-Fourier transform

Input: Two blurred images $\mathcal{G}_1$ and $\mathcal{G}_2$.

Output: The deblurred image $\mathcal{F}$.

Begin:

1. Calculate the size of $\mathcal{G}_1$ and $\mathcal{G}_2$, which is $M \times N$.
2. Calculate the $\mathbb{G}_1$ and $\mathbb{G}_2$, which are the 2D z-Fourier transform of $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively.
3. Define the random number of pairs of i^{th} rows or j^{th} columns of $\mathbb{G}_1$ and $\mathbb{G}_2$, which is the trial number for the calculation of the degree of AGCD
4. Initialise the array that stores the result of the calculation of the degree of AGCD. The size of the array depends on the random number of trials on rows or columns.
5. **while** number of trials **do**
6. Select the two polynomials $f(x)$ and $g(x)$ which have a pair of i^{th} rows or j^{th} columns from $\mathbb{G}_1$ and $\mathbb{G}_2$.
7. Pre-process $f(x)$ and $g(x)$ to be the polynomials $\tilde{f}_{\theta_0}(y)$ and $\alpha_0\tilde{g}_{\theta_0}(y)$.
8. Calculate the value of t for the trial using the QR decomposition of the Sylvester resultant matrix and its sub-resultant matrix $S(\tilde{f}_{\theta_0}, \alpha_0\tilde{g}_{\theta_0})$ using (3.32).
9. Save the result into an array that has been defined for the calculation of the degree of AGCD.
10. **end**
11. The computed t is equal to the mode of the entries in the array that stores the result of the calculation of the degree of AGCD.
12. Calculate the M univariate AGCD for each row of $\mathbb{G}_1$ with the associated row of $\mathbb{G}_2$.

    For $i = 1 : M$

```
    (a) Define the $i$th row vector $r_1 = \mathbb{G}_1(i,:)$ and $r_2 = \mathbb{G}_2(i,:)$
    (b) Pre-process the coefficients of the two inexact polynomials $r_1$ and $r_2$ by:
        (I) Normalise the coefficient by the geometric mean $\hat{r}_1$ and $\hat{r}_2$.
        (II) Compute the optimal value of $\alpha_0$ and $\theta_0$ by using the linear programming problem $(LP)$.
    (c) Compute the coefficient $d(i)$ of AGCD$(\hat{r}_{1\theta_0}, \alpha_0\hat{r}_{2\theta_0})$ by using Approximate Polynomial Factorisation APF.
    (d) Embed the $i$th row of the coefficient of AGCD $d(i)$ in the matrix $A$.
        $A(i,:) = d(i)$
    **End For**
13. Calculate the $N$ univariate AGCD for each column of $\mathbb{G}_1$ with the associated column of $\mathbb{G}_2$.
    **For** $j = 1 : N$
    (a) Define the $j$th column vector $c_1 = G_1(:,j)$ and $c_2 = G_2(:,j)$
    (b) Pre-process the coefficients of the two inexact polynomials $c_1$ and $c_2$ by:
        (I) Normalise the coefficient by the geometric mean $\hat{c}_1$ and $\hat{c}_2$.
        (II) Compute the optimal value of $\alpha_0$ and $\theta_0$ by using the linear programming problem $(LP)$.
    (c) Compute the coefficient $d(j)$ of AGCD$(\hat{c}_{1\theta_0}, \alpha_0\hat{c}_{2\theta_0})$ by using Approximate Polynomial Factorisation APF.
    (d) Embed the $j$th column of the coefficient of AGCD $d(j)$ in the matrix $B$.
        $B(:,j) = d(j)$
    **End For**
14. solve (5.16) using the singular value decomposition SVD of the coefficient matrix in (5.16) to calculate the solution of $a(i), i = 1, \ldots, M$ and $b(j), n = 1, \ldots, N$.
15. Compute the recovered true images in $z$-Fourier transform $\mathbb{F}$ using (6.26).
```

16. Apply the inverse z-Fourier transform to the $\mathbb{F}$ to compute the recovered true image $\mathcal{F}$.

End

Algorithm 6.4 is used in the following examples to recover the true images.

6.4.1 Example

This section contains three examples in which the distorted image is restored using Algorithm 6.4. Raw images of 180×180 pixels have been used to examine the developed method. The effect of noise and non-separable PSF on image restoration is examined in these examples.

The examples show that the use of the method of AGCD by the APF technique resulted in a better version estimation of the exact image compared with other linear algebra methods described in [1, 2]. Additionally, the examples show that the method that used the Bezout matrix [1, 2] is much more sensitive to the noise than the developed methods that use the Sylvester resultant matrix.

Example 6.1. *In this example, an exact image is blurred by non-separable Gaussian blurring PSF $\mathcal{H}_1$ and non-separable motion blurring PSF $\mathcal{H}_2$ of size 27×17 pixels (see Figure 6.3) in order to generate the two distorted versions of the images. Each of the non-separable PSFs and images have an error $\mathcal{E} = 0$ and additive noise $\mathcal{N} = 0$, respectively.*

The distorted images are then processed using Algorithm 6.4 and another linear algebra method developed by Li et al. [1, 2]. The results of each of the processing methods are given in Figure 6.5, with Figure 6.4 showing the computed value of t using (3.32). The most frequently computed degree (the mode) is 179 for rows and columns, and the size of the deblurred image is therefore 180×180.

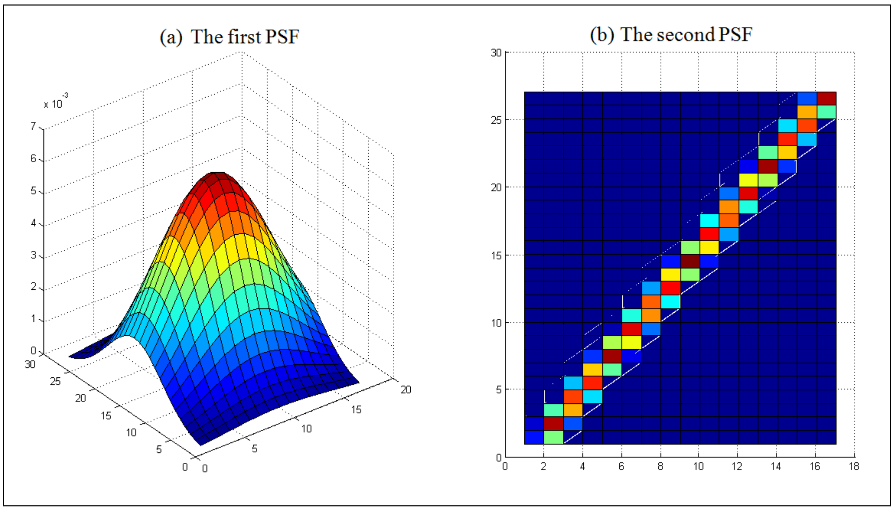


FIGURE 6.3: The (a) first and (b) second PSFs that are applied to the two distorted versions of the true images, respectively, in Example 6.1.

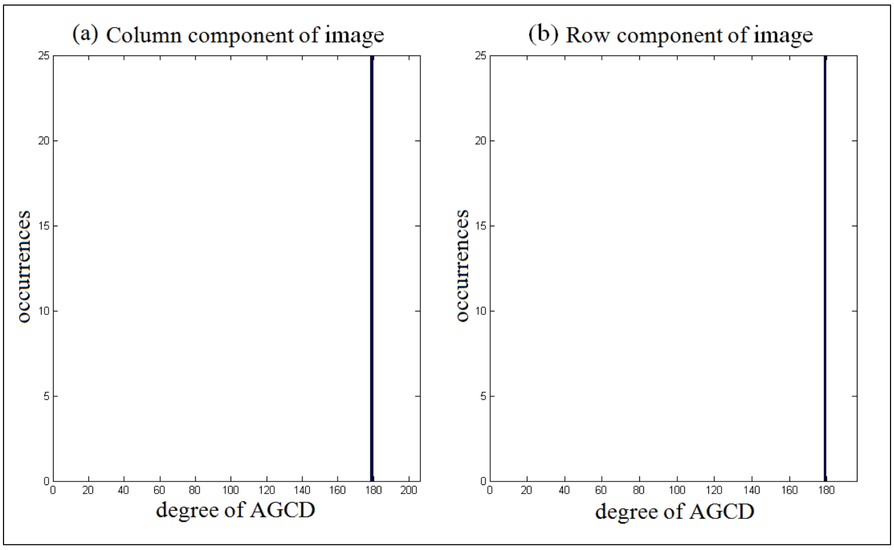


FIGURE 6.4: The histogram of the degree of AGCD between two polynomials in rows and columns in the absence of noise for Example 6.1.

TABLE 6.1: computing the degree of AGCD between two polynomials in rows and columns in the absence of noise for Example 6.1.

	Mode	Median	Mean
Computing Rank in Rows	179	179	179
Computing Rank in Columns	179	179	179

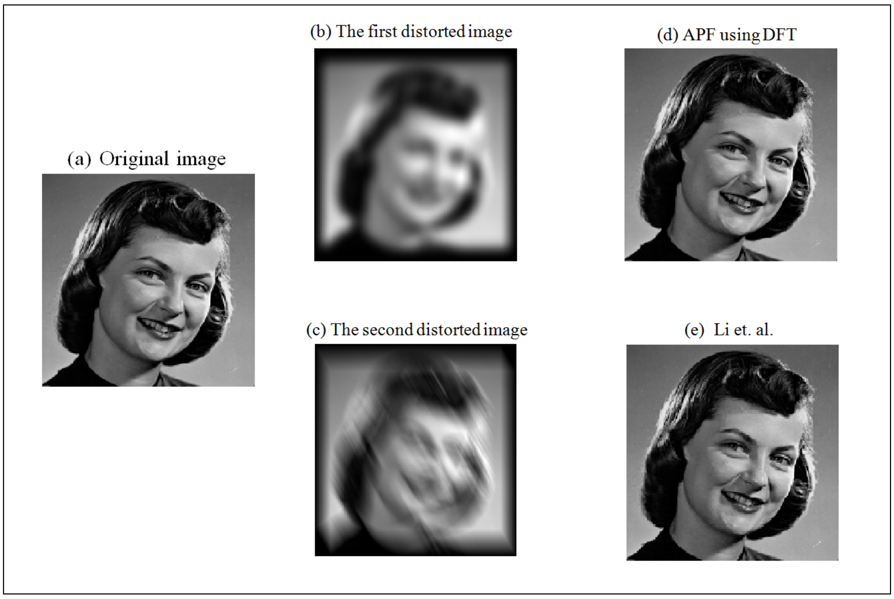


FIGURE 6.5: Blind Image Deconvolution using (d) the developed method and (e) the method developed by Li et al. [1, 2], for Example 6.1.

Then another two blurred images were generated from the true image using the same PSFs in Figure 6.3. But in this case, each of the PSFs has an error $\mathcal{E} = 1 \times 10^{-7}$ and then noise $\mathcal{N} = 1 \times 10^{-8}$ is added to the result of each of the convolutions.

The distorted images are then processed similarly using Algorithm 6.4 and the method developed by Li et al. [1, 2]. The results of each of the processing methods, in the presence of noise, are given in Figure 6.7, whereas Figure 6.6 shows the computed value of t and the most frequently computed degree (the mode) is 179 for rows and columns, and the size of the deblurred image is therefore 180×180, which is correct.

TABLE 6.2: computing the degree of AGCD between two polynomials in rows and columns in the presence of noise for Example 6.1.

	Mode	Median	Mean
Computing Rank in Rows	179	179	179.6
Computing Rank in Columns	179	179	171.88

Table 6.3 gives a comparison of the relative errors produced with the BID method that is based on linear algebra. It clearly shows that the developed method better estimates the exact image.

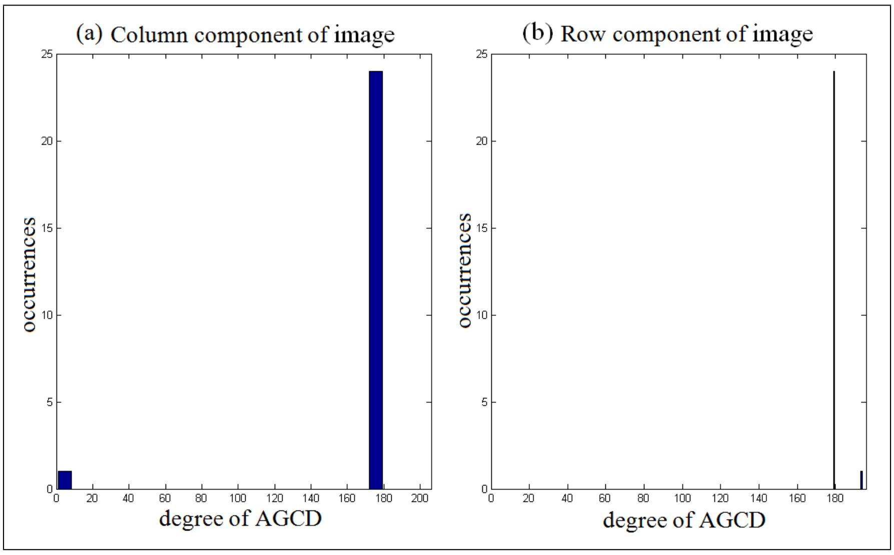


Figure 6.6: The histogram of the degree of AGCD between two polynomials in rows and columns in the presence of noise for Example 6.1.

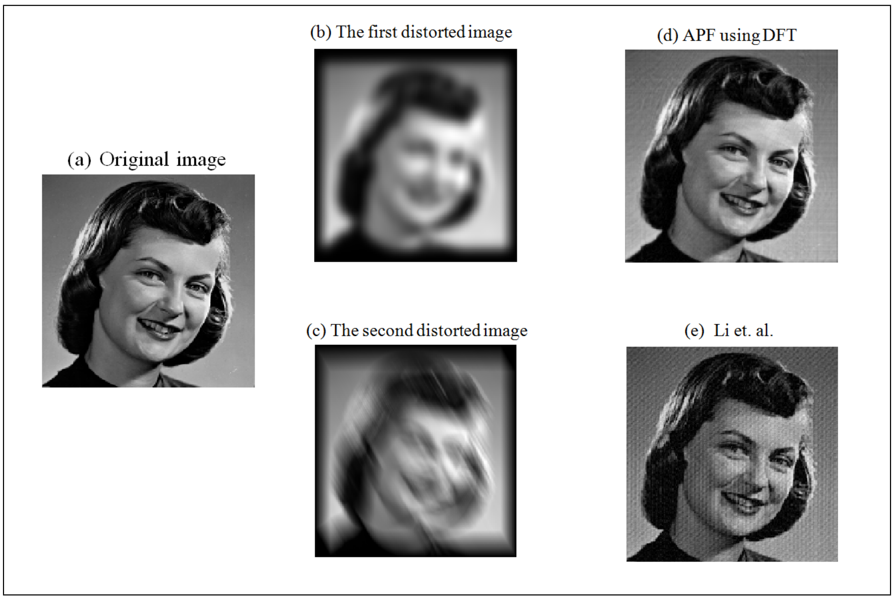


Figure 6.7: Blind Image Deconvolution of a noisy blurred image using (d) the developed method and (e) the method developed by Li et al. [1, 2], for Example 6.1.

Table 6.3: Comparison table for the distorted image in Example 6.1.

BID Methods	Additive noise	Size of PSF	SNR	MSE	Relative error
APF	absent	not specified	172.21	6.008×10^{-18}	5.13×10^{-9}
APF	present	not specified	28.68	0.0014	0.0605
Li et al.	absent	specified	43.68	4.282×10^{-5}	0.0147
Li et al.	present	specified	23.898	0.0041	0.1079

Example 6.2. *In this example, two non-separable PSFs are generated as a result of the compilation of separable Gaussian blurring and non-separable motion blur. These two non-separable PSFs of size* 19×11 *pixels are shown in Figure 6.8. An exact image is processed by the generated non-separable PSFs. Each of the non-separable PSFs and images have an error* $\mathcal{E} = 0$ *and additive noise* $\mathcal{N} = 0$*, respectively.*

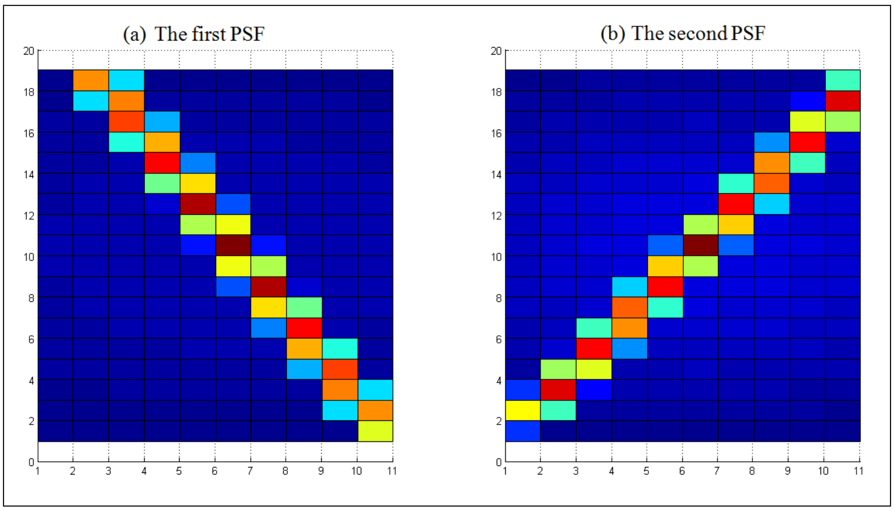


Figure 6.8: The (a) first and (b) second PSFs that are applied to the two distorted versions of the true images, respectively, in Example 6.2.

In the absence of noise, the distorted images are processed using Algorithm 6.4 and another linear algebra method developed by Li et al. [1, 2]. The results of each of the processing methods are given in Figure 6.10, where Figure 6.9 shows the computed value of t using (3.32). The most frequently computed degree (the mode) is 179 *for rows and columns, and the size of the deblurred image is therefore* 180×180.

Table 6.4: computing the degree of AGCD between two polynomials in rows and columns in the absence of noise for Example 6.2.

	Mode	Median	Mean
Computing Rank in Rows	179	179	179
Computing Rank in Columns	179	179	179

Then another two blurred versions of the image were generated using the same PSFs in Figure 6.8. But in this case, each of PSF has an error $\mathcal{E} = 1 \times 10^{-6}$ *and then noise* $\mathcal{N} = 1 \times 10^{-7}$ *is added to the result of each of the convolutions.*

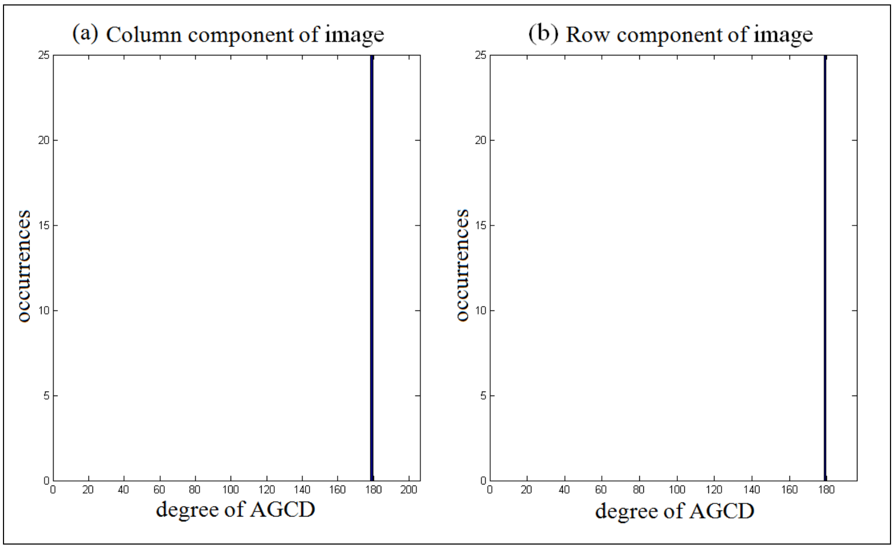


FIGURE 6.9: The histogram of the degree of AGCD between two polynomials in rows and columns in the absence of noise for Example 6.2

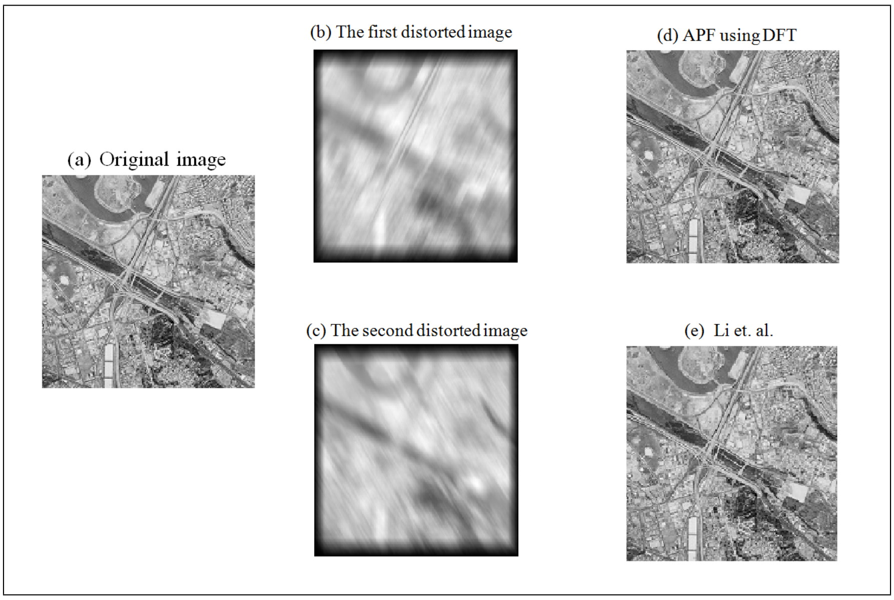


FIGURE 6.10: Blind Image Deconvolution using (d) the developed method and (e) the method developed by Li et al. [1, 2], for Example 6.2.

The distorted images are then processed similarly using Algorithm 6.4 and the method developed by Li et al. [1, 2]. The results of each of the processing methods, in the presence of noise, are given in Figure 6.12, whereas Figure 6.11 shows the computed value of t*, and the most frequently computed degree (the mode) is* 179 *for rows and columns, and the size of the deblurred image is therefore* 180×180*, which is correct.*

Table 6.6 gives a comparison of the relative errors produced with the BID method based on linear algebra. It clearly shows that the developed method better estimates the exact

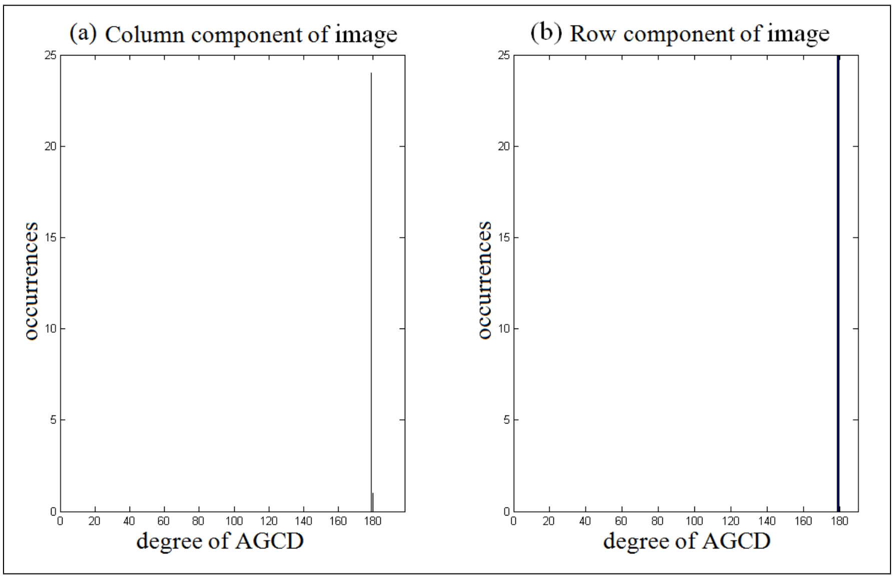


FIGURE 6.11: The histogram of the degree of AGCD between two polynomials in rows and columns in the presence of noise for Example 6.2

TABLE 6.5: computing the degree of AGCD between two polynomials in rows and columns in the presence of noise for Example 6.2.

	Mode	Median	Mean
Computing Rank in Rows	179	179	179
Computing Rank in Columns	179	179	179.04

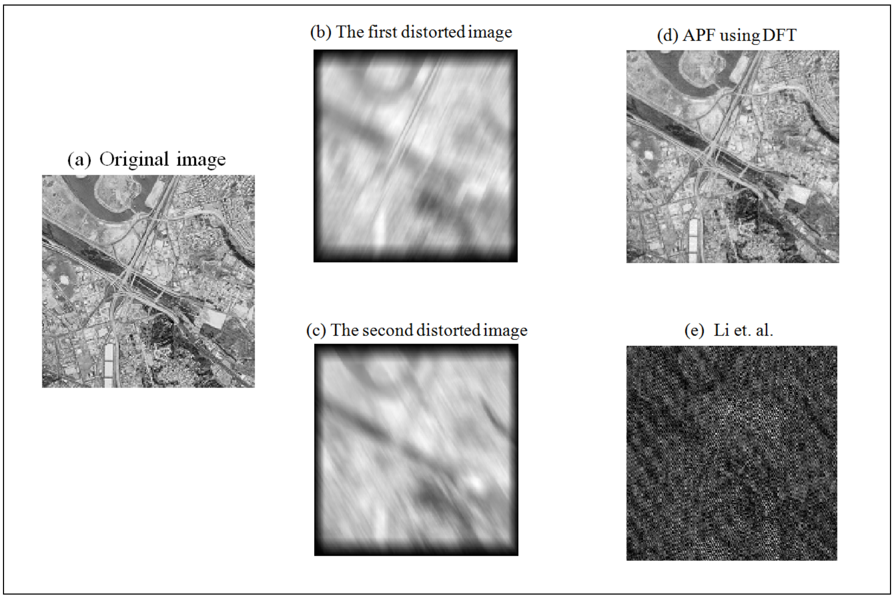


FIGURE 6.12: Blind Image Deconvolution of a noisy blurred image using (d) the developed method and (e) the method developed by Li et al. [1, 2], for Example 6.2.

image. Furthermore, the method developed by Li et al. [1, 2] fails to recover the true image in the presence of noise and measurement error.

Table 6.6: Comparison table for the distorted image in Example 6.2.

BID Methods	Additive noise	Size of PSF	SNR	MSE	Relative error
APF	absent	not specified	228.71	1.35×10^{-23}	5.39×10^{-12}
APF	present	not specified	27.29	0.0019	0.0707
Li et al.	absent	specified	52.92	5.099×10^{-6}	0.0034
Li et al.	present	specified	8.1396	0.1535	0.5852

Example 6.3. *In this example, two non-separable PSFs are generated as a result of the compilation of separable Gaussian blurring and non-separable motion blur. These two non-separable PSFs of size* 39×33 *pixels are shown in Figure 6.13. An exact image is processed by generating non-separable PSFs. Each of the non-separable PSFs and images have an error* $\mathcal{E} = 0$ *and additive noise* $\mathcal{N} = 0$*, respectively.*

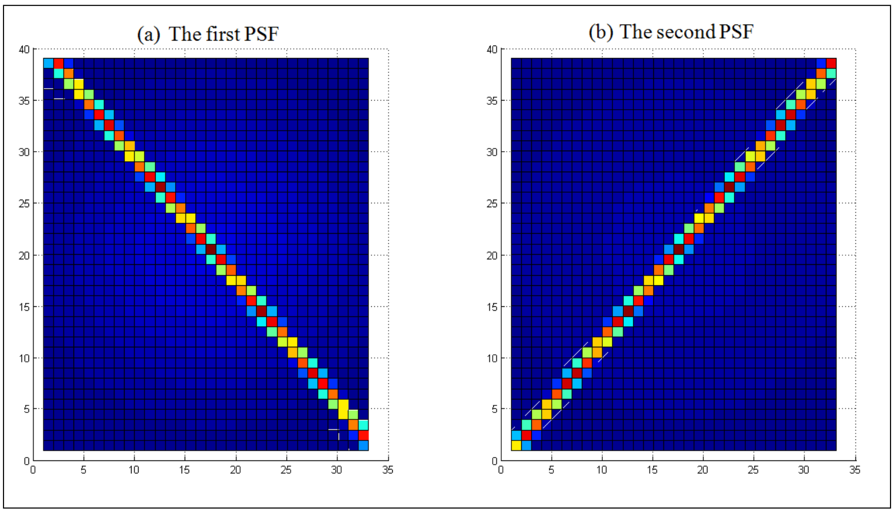


Figure 6.13: The (a) first and (b) the second PSFs that are applied to the two distorted versions of the true images, respectively, in Example 6.3.

In the absence of noise, the distorted images are processed using Algorithm 6.4 and another linear algebra method developed by Li et al. [1, 2]. The results of each of the processing methods are given in Figures 6.15, where Figure 6.14 shows the computed value of t *using (3.32). The most frequently computed degree (the mode) is* 179 *for rows and columns, and the size of the deblurred image is therefore* 180×180.

Table 6.7: computing the degree of AGCD between two polynomials in rows and columns in the absence of noise for Example 6.3.

	Mode	Median	Mean
Computing Rank in Rows	179	179	179
Computing Rank in Columns	179	179	179

Then another two blurred versions of the image were generated using the same PSFs in Figure 6.13. But in this case, each of the PSF has an error $\mathcal{E} = 1 \times 10^{-7}$ *and then noise* $\mathcal{N} = 1 \times 10^{-8}$ *is added to the result of each of the convolutions.*

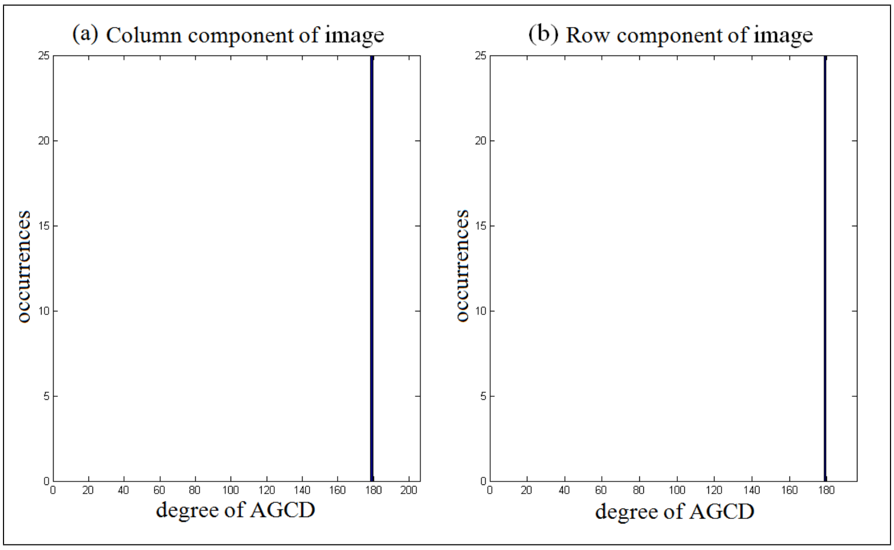


FIGURE 6.14: The histogram of the degree of AGCD between two polynomials in rows and columns in the absence of noise for Example 6.3.

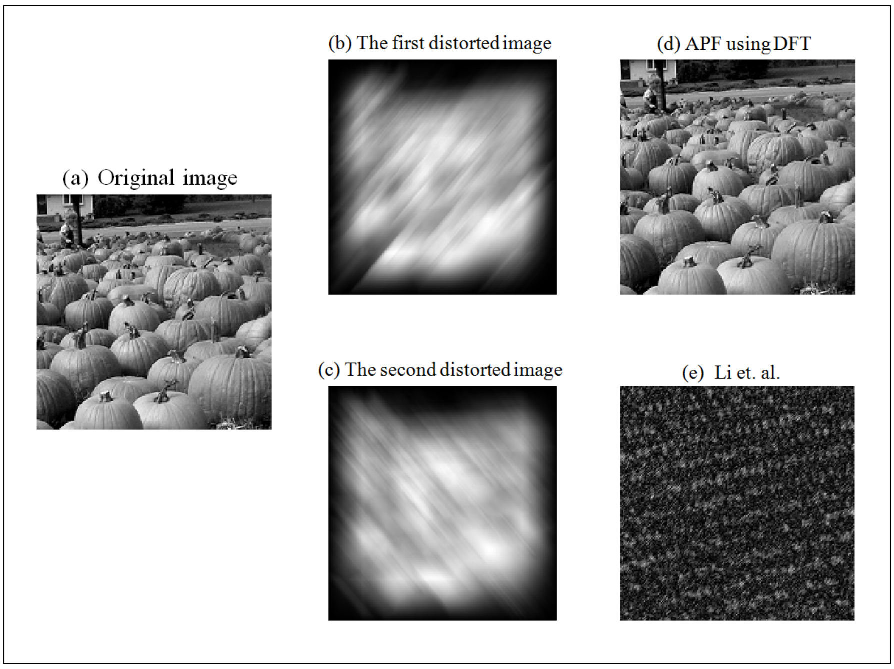


FIGURE 6.15: Blind Image Deconvolution using (d) the developed method and (e) the method developed by Li et al. [1, 2], for Example 6.3.

The distorted images are then processed similarly using Algorithm 6.4 and the method developed by Li et al. [1, 2]. The results of each of the processing methods, in the presence of noise, are given in Figure 6.17, whereas Figure 6.3 shows that the most frequently computed degree (the mode) is 179 *for rows and columns, and the size of the deblurred image is therefore* 180×180, *which is correct.*

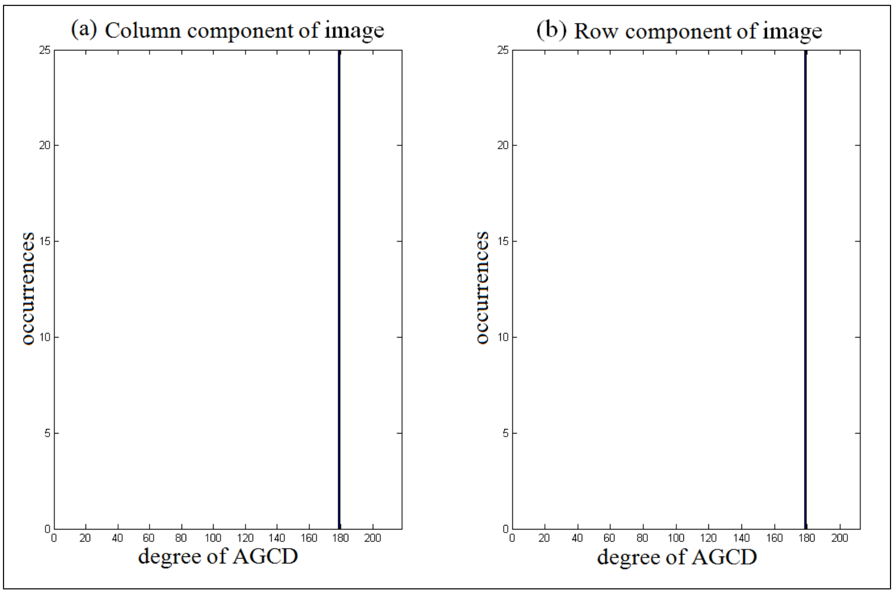


FIGURE 6.16: The histogram of the degree of AGCD between two polynomials in rows and columns in the presence of noise for Example 6.3

TABLE 6.8: computing the degree of AGCD between two polynomials in rows and columns in the presence of noise for Example 6.3.

	Mode	Median	Mean
Computing Rank in Rows	179	179	179
Computing Rank in Columns	179	179	179

Table 6.9 gives a comparison of the relative errors produced with the BID method that is based on linear algebra. It clearly shows that the developed method better estimates the exact image. Furthermore, the method developed by Li et al. [1, 2] fails to recover the true image in a large amount of non-spreadable PSF, even in the absence of noise and measurement error.

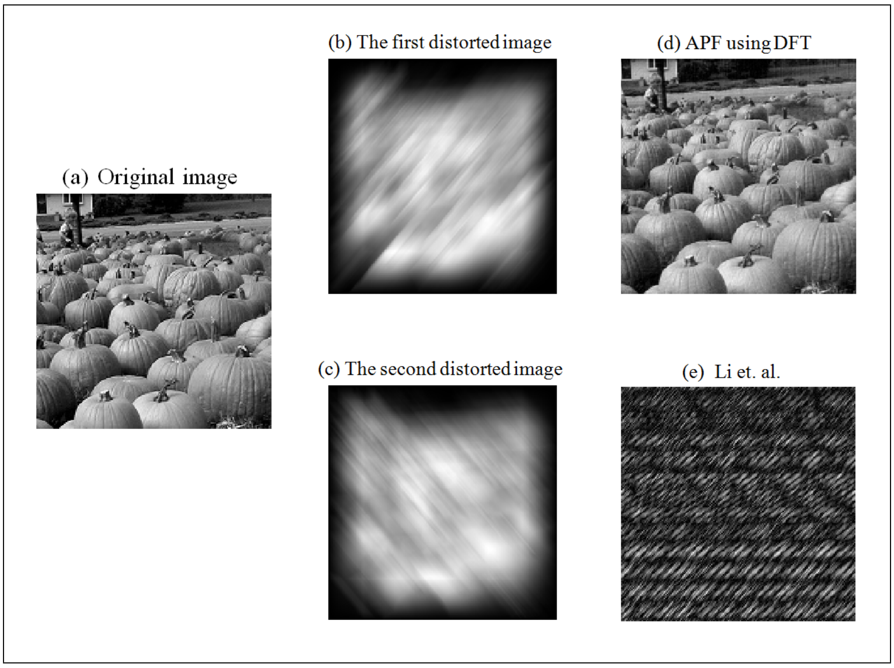


Figure 6.17: Blind Image Deconvolution of a noisy blurred image using (d) the developed method and (e) the method developed by Li et al. [1, 2], for Example 6.3.

Table 6.9: Comparison table for the distorted image in Example 6.3.

BID Methods	Additive noise	Size of PSF	SNR	MSE	Relative error
APF	absent	not specified	210.82	8.27×10^{-22}	5.34×10^{-11}
APF	present	not specified	29.125	0.0012	0.0656
Li et al.	absent	specified	8.7567	0.1331	0.7822
Li et al.	present	specified	9.1345	0.1221	0.7713

6.5 Summary

This chapter has considered the application of a BID solution of non-separable blur using the structure matrix and z-Fourier transform. The true image is calculated as the AGCD of two blurred images, in which two blurred images of the same scene are given. Chapter 5 showed the developed BID method for a non-separable PSF, and it was shown that it can be calculated from the AGCD of two different blurred images. The cases considered in Chapter 5 used non-complex non-separable blurring functions. Therefore, This chapter was found that the frequency domain implementation of the method is more suitable and applicable for non-separable PSF estimation, in general, and complex structure non-separable function, in particular.

The implementation of the developed method in the frequency domain is presented in this chapter. The developed frequency domain algorithm has been applied on non-separable blur of different sizes. The implementation of the method in the frequency domain solved the non-separable case of the PSF with better estimation results. The results of the method is compared to another algebraic deblurring method, Li et al. [1, 2]. The developed method showed better estimation of the true image than the Li et al. method. Algebraic methods in general are sensitive to noise. With low level of noise, the experiments showed that Li et al. method failed to recover the image whereas the developed method in this research was able to estimate the image well. The level of noise used in the experiments is low in relation to statistical BID methods, however it is relatively high level in algebraic BID methods. In comparison to the spatial implementation of the method presented in Chapter 5, the frequency domain implementation of the developed method is able to estimate the true images with non-separable PSF of any complexity.

Chapter 7

Developed Method Performance and Discussion

7.1 Introduction

This research study focused on the development of a solution for the blind image deconvolution problem, specifically, blurred images that are influenced by spatially invariant blur. The developed BID method is designed to be robust and computationally efficient in the estimation of the PSF. This is achieved by implementing an AGCD computation method using APF, which is considered in Chapters 4 and 5. This chapter will analyse and discuss in more detail the performance of the developed method and compare the results to the state-of-the-art methods. It examines and compare the performance of the BID methods, as applied to artificially blurred images, in recovering the original image by considering different levels of additive noise, measurement errors and the size of the PSF. The results show that this method as applied to artificially blurred images has a better performance than the state-of-the-art methods (discussed in Chapter 2). The application of the developed method to real blurred images without pre-processing failed consistently as discussed in Section 7.3. This chapter will start by analysis of four image deconvolution experiments.

7.2 Experiments and Discussions

Four experiments have been carried out, using MATLAB. Experiment one examines the performance of the BID methods, including the developed method, in relation to changes in PSF width and additive noise. Experiment two examines the behaviour of the state-of-the-art methods in recovering the true image from a blurred image with an unknown PSF size. The third experiment examines the performance of the developed method against other state-of-the-art linear algebra methods. Images with different additive noise levels and errors in PSF were used in this experiment. In all three experiments, an image size of 180×180 is used. Lastly, experiment four examines the performance of the deblurring algorithms for feature detection. Face feature detection and circle feature detection experiments are performed on deblurred images from various deblurring algorithms.

7.2.1 Experiment One: BID performance with variable PSF width and additive noise

The true images, aerial map of San Diego, Camera man, girl face and grass(see Figure 7.1), are used to generate blurred images with PSFs of different size and different levels of additive noise. The blurred images are then processed by nine deblurring methods. These methods are grouped into three sets. Set 1 consists of four basic deblurring methods: Richardson-Lucy algorithm [6], maximum likelihood algorithm (ML) [7], Wiener algorithm [8] and the regularisation method [9]. These are implemented in the image processing toolbox in the MATLAB. Set 2 consists of three methods based on Bayesian statistics theory: the methods of Shearer et al. [4], Babacan et al. [5] and Perrone and Favaro [3]. The third set consists of two methods based on linear algebra, specifically AGCD; these are the APF methods developed in this research study and method of Li et al. [1, 2].

Forty-two blurred images were generated from each true images. Each of these forty-two blurred images is generated from a unique PSF and noise level. The following levels of

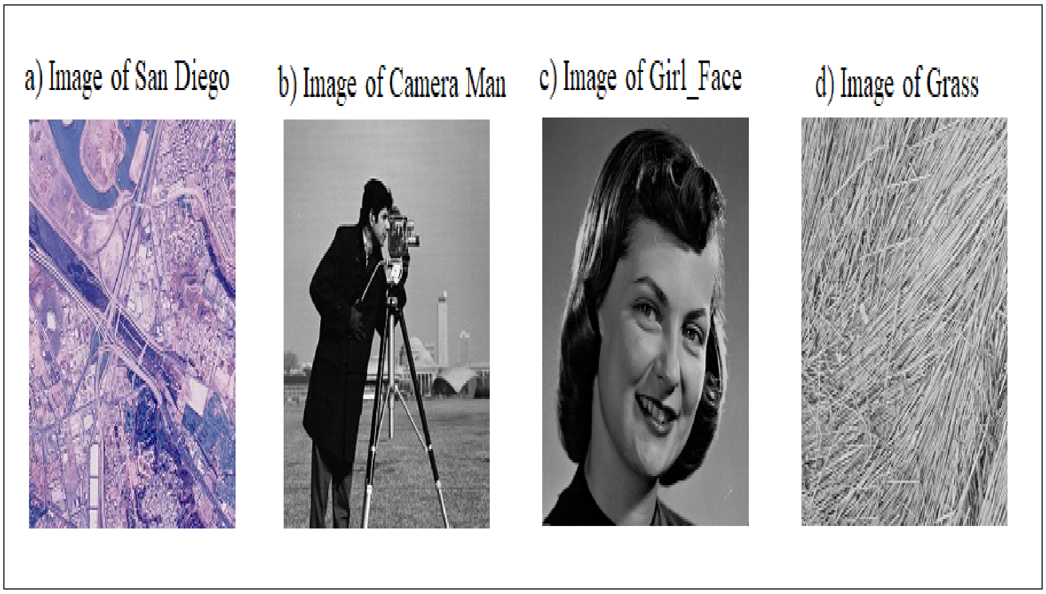


FIGURE 7.1: The true images that are used for experiment one

noise and PSF sizes are used: PSF sizes: $5, 15, 25, 35, 45, 55$.

Noise levels: $N = 10^{-11}, 10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}$.

The blurring model used in the generation of the blurred images is given by the following formula:

$$\mathcal{G} = \mathcal{H} \otimes \mathcal{F} + \mathcal{N}, \tag{7.1}$$

The generated blurred images are then processed with each of the above methods, in turn, to produce the deblurred images. The deblurred images are then compared to the original image using the performance criterion, specifically, Relative Error. The error results are given in Figures 7.2, 7.6 and 7.10.

Figures 7.2, 7.3, 7.4 and 7.5 shows the error results of the four basic methods that are implemented in the image processing toolbox in the MATLAB. The minimum error obtained from this set of methods is of the same order of magnitude. The minimum error ranged from 0.0448 to 0.0905, corresponding to zero additive noise and the smallest PSF size, which equals 5×5. The maximum error ranged from 0.2440 to 0.3859, corresponding to the largest noise level added, which is $\varepsilon = 10^{-6}$, and the largest PSF size, which equals 55×55.

The figures show a jump in the error value when the PSF size increased from the smallest PSF size, 5×5, to a PSF size of 15×15. In addition, the additive error level did not affect the error results in any significant way. This shows that, in the previous experiments, decreasing the noise level for a particular PSF does not improve the quality of the deblurred image. In other words, the quality of the deblurred images did not improve, regardless of the additive noise level present in the blurred images. This also shows that the size of the PSF is an effective factor that has a considerable impact on the quality of the produced images. The error results for this set of methods are high compared to the results of the developed method discussed later.

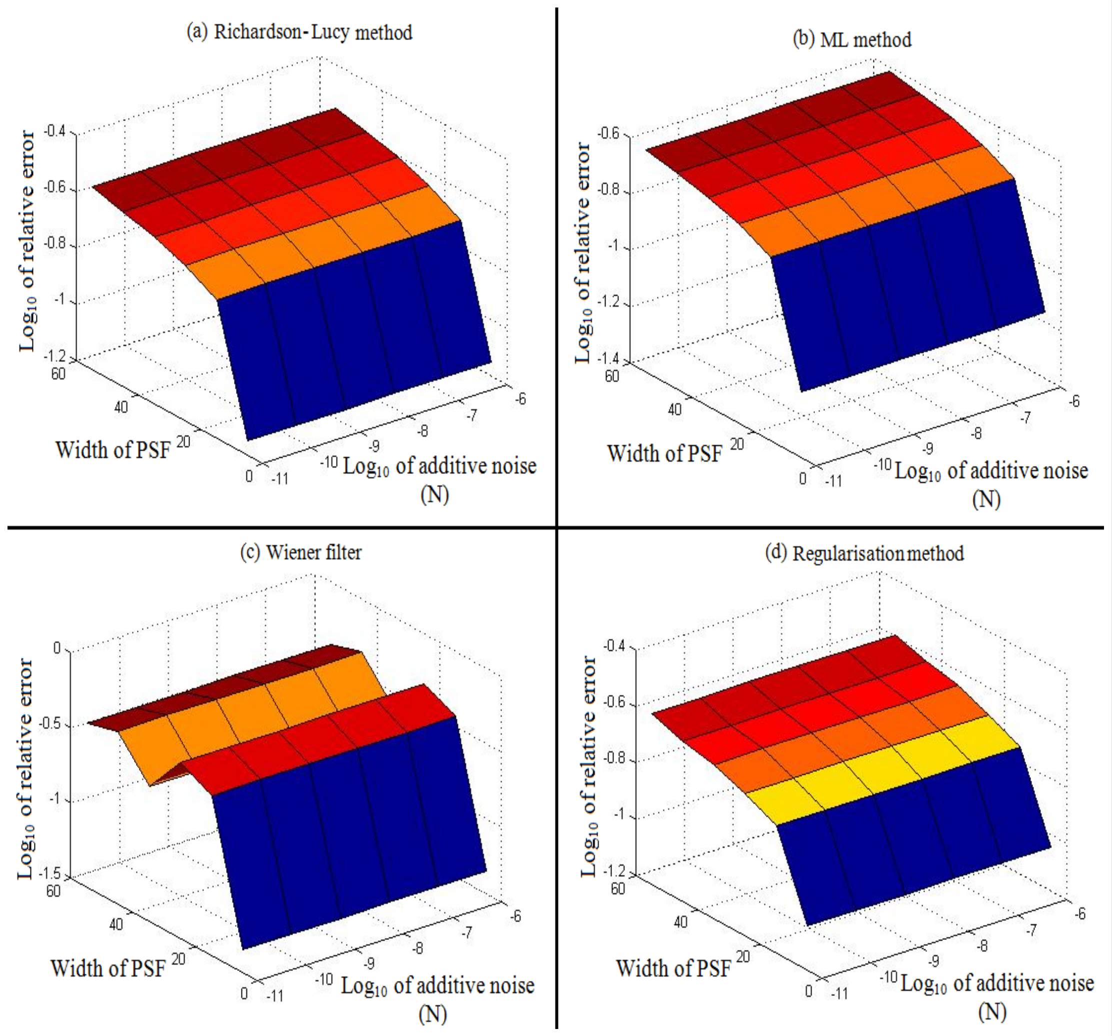


FIGURE 7.2: Basic deblurring methods error results of experiment one when a true image of San Diega has used

The second set of results, from Bayesian-based methods, are presented in Figures 7.6,

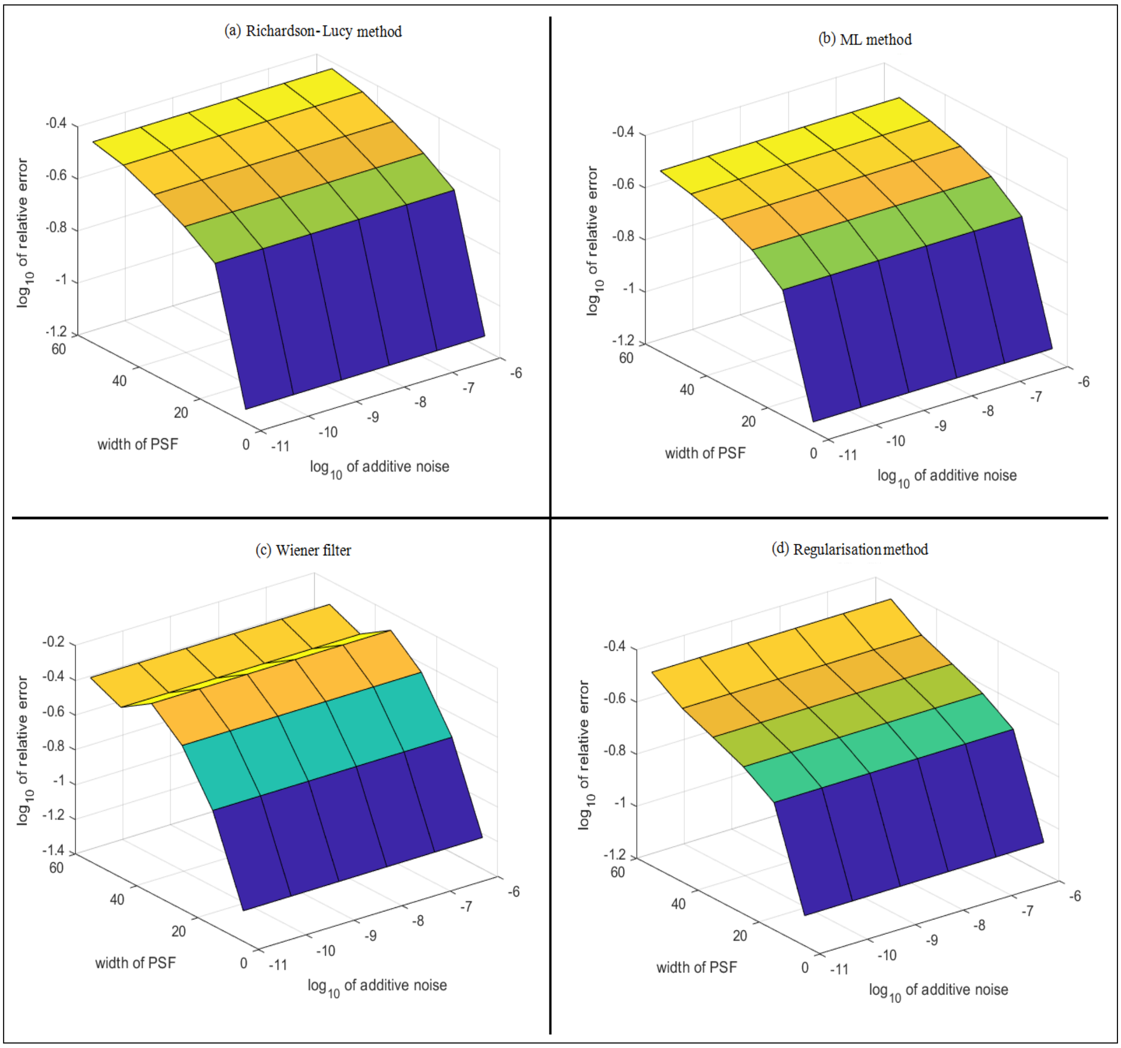


FIGURE 7.3: Basic deblurring methods error results of experiment one a true image of camera man has used

7.7, 7.8 and 7.9. The minimum recorded error for this set of methods is again of the same order of magnitude. The lowest error value recorded for this set is, however, higher than the lowest error recorded for the basic methods in the first set of results. The minimum error ranged from 0.1244 to 0.1541, corresponding to zero additive noise and the smallest PSF size, which equals 5×5. The maximum error ranged from 0.2576 to 0.3166, corresponding to the largest added noise level, which is $\varepsilon = 10^{-6}$, and the largest PSF size, which equals 55×55. The highest level of error for this set is similar to the results of set one.

The figures show a gradual increase in the error as the size of the PSF is increased. The resulting image error increased by about two times the lowest error value in the

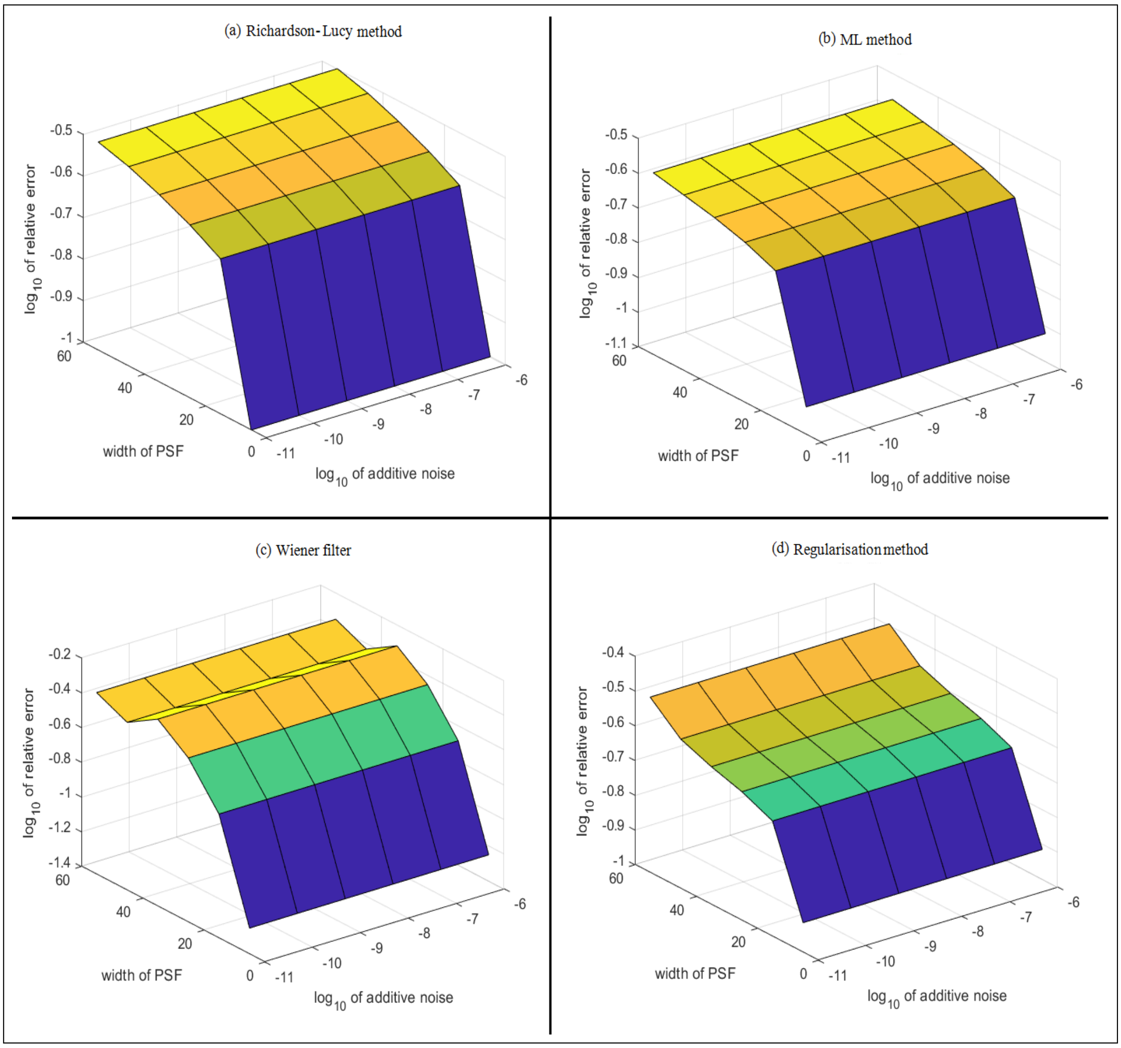


FIGURE 7.4: Basic deblurring methods error results of experiment one a true image of grass has used

worst case of a blurred image. Similar to the basic deblurring methods, the error in the resulting image did not improve as the additive noise decreased to zero.

The third set of error results, from linear algebra-based methods, are presented in Figures 7.10, 7.11, 7.12, and 7.13. The minimum error values are 4.77×10^{-13} for method of Li et al. [1, 2] and 1.16×10^{-13} for the developed APF method. These error levels are of the order of 10^{-10} times than the error results of the set one and set two deblurring methods. Method of Li et al. [1, 2] shows a jump in the error value when the PSF size increased from the smallest PSF size of 5×5 to a PSF size of 15×15. The error level increased considerably for this size of PSF and remained constant for higher PSF sizes. The image quality improved with the decrease in the additive error level. However, in

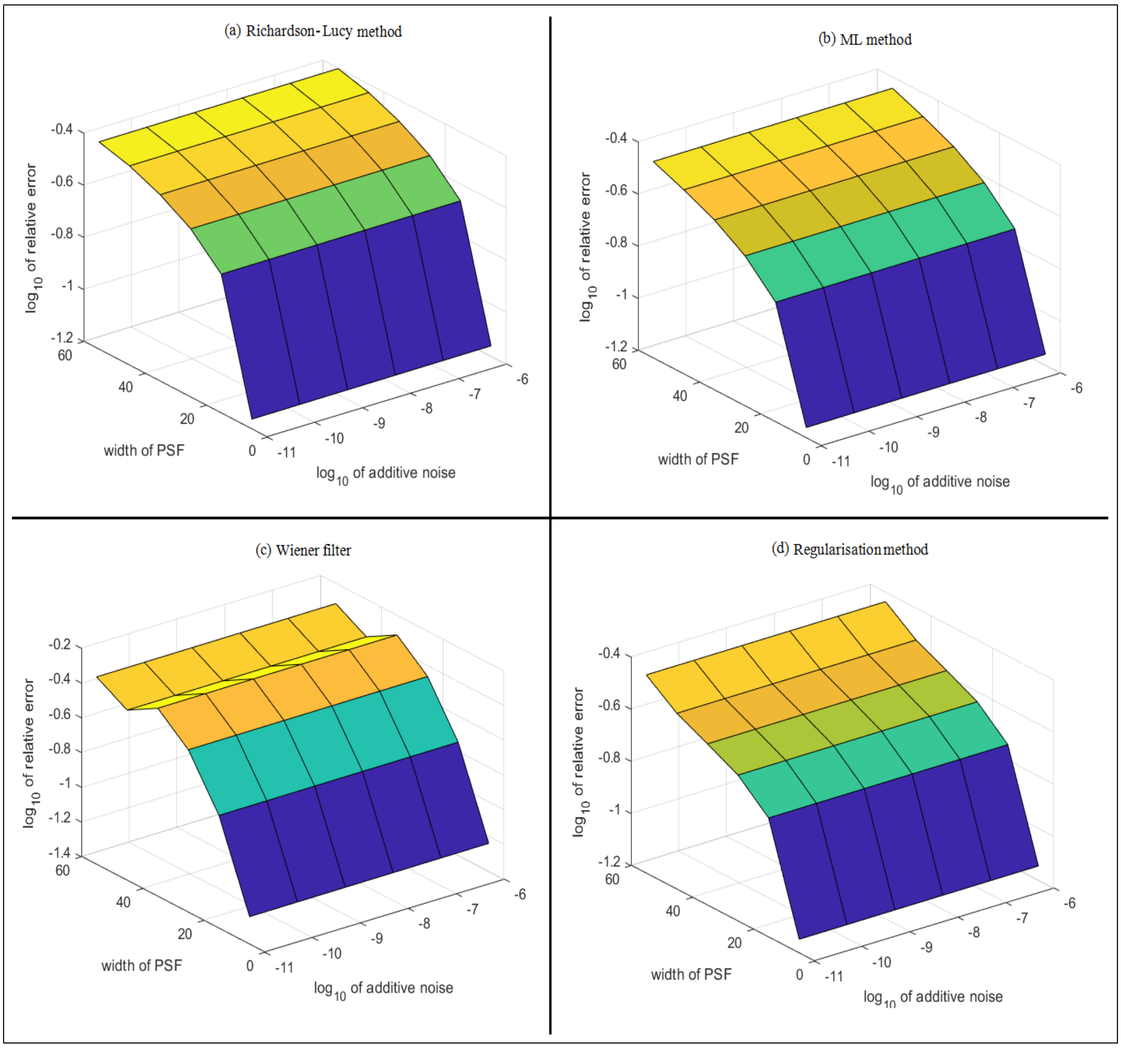


FIGURE 7.5: Basic deblurring methods error results of experiment one a true image of girl face has used

comparison to the error results from set one and set two methods, the error results of the method of Li et al. [1, 2] are lower and approximately of the same order of magnitude for high PSF sizes.

Figure 7.10, 7.11, 7.12, and 7.13 shows the error results of the developed APF method. The error results are considerably smaller than that of set one and two and the method of Li et al. [1, 2]. The increase in the size of the PSF did not affect the error result values and it remained almost as low as in the free of additive noise and smallest size PSF cases. In these experiments, the developed method had a better performance than all the other methods, in particular, it determines the correct size and elements of the PSF and then uses it in recovering the original image. In addition, this method, as noted

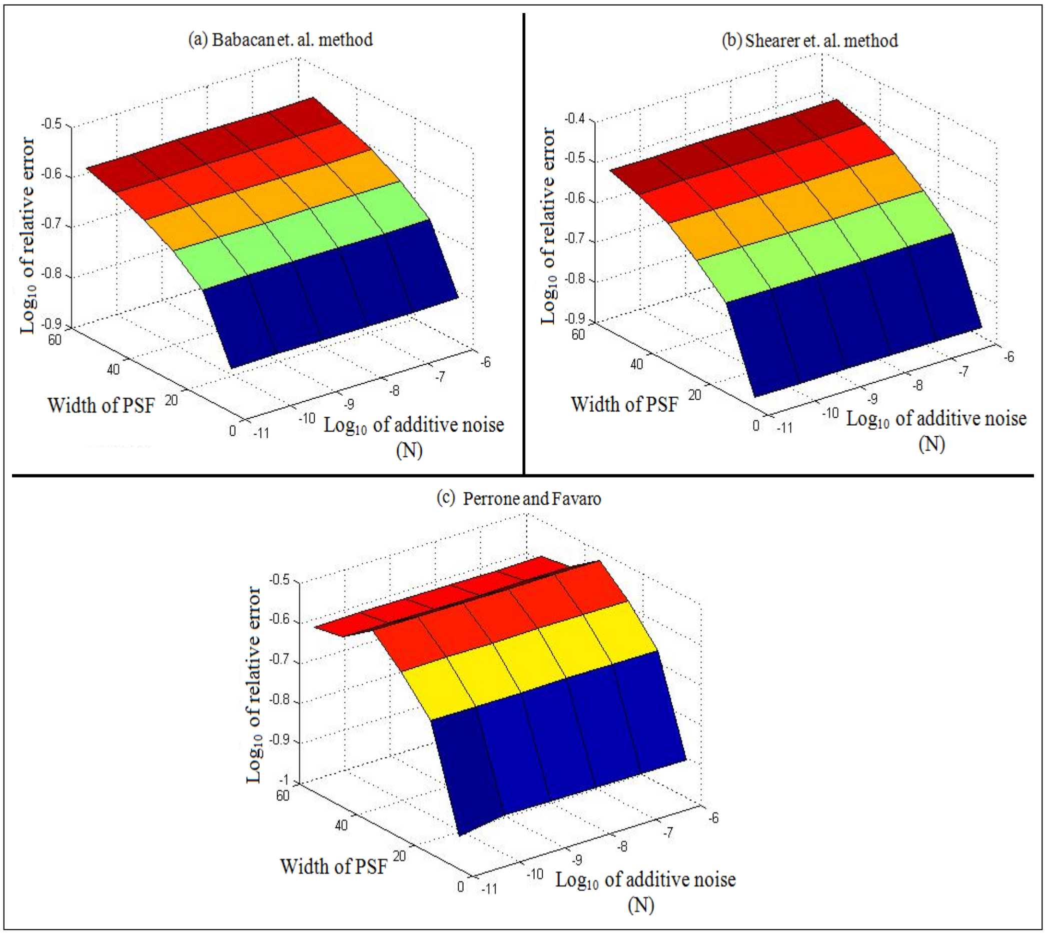


FIGURE 7.6: Bayesian statistical theory based methods error results of experiment one when a true image of San Diega has used

above, does not require any prior knowledge about the PSF, which is a prerequisite for all the other methods. The additive noise level did affect the resultant error in the recovered image. It is expected that as the noise level increases, the error in the deblurred image increases; however, the error results of this method for the worst considered case are still lower than that of all other methods. In this research study, the noise level has been taken into account during the calculation of the PSF, as discussed in Chapters 4 and 5. Regardless of the additive noise level present in the blurred image, the developed algorithm is able to compute the correct PSF. The additive noise level should be taken into consideration during the deconvolution stage in order to recover a better quality image from the blurred image. In this research study the noise level was not taken into account in the deconvolution process and will be considered in future work. It is expected that, by taking into account the additive noise in the deconvolution stage, the

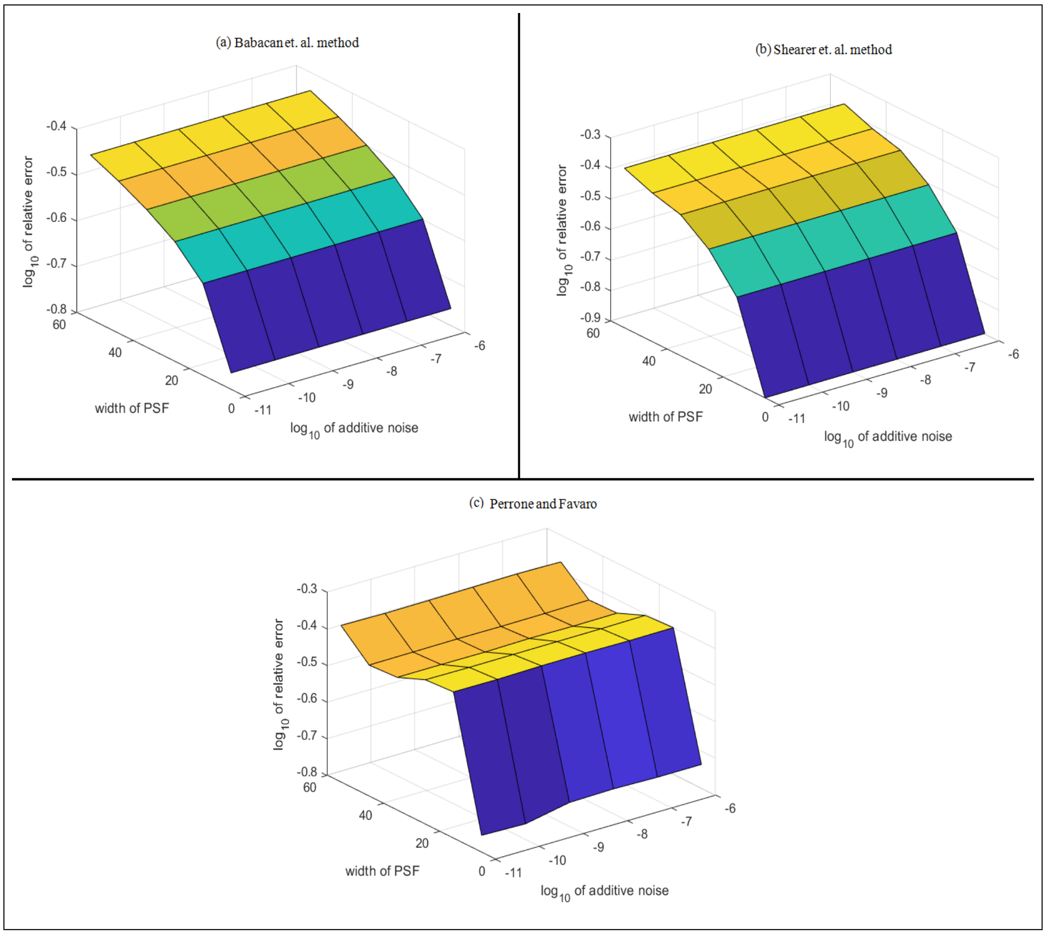


FIGURE 7.7: Bayesian statistical theory based methods error results of experiment one when a true image of camera man has used

resultant error in the recovered image will be much lower than in the currently obtained results. It is also expected that the error values will increase at a much smaller rate as the noise level increases.

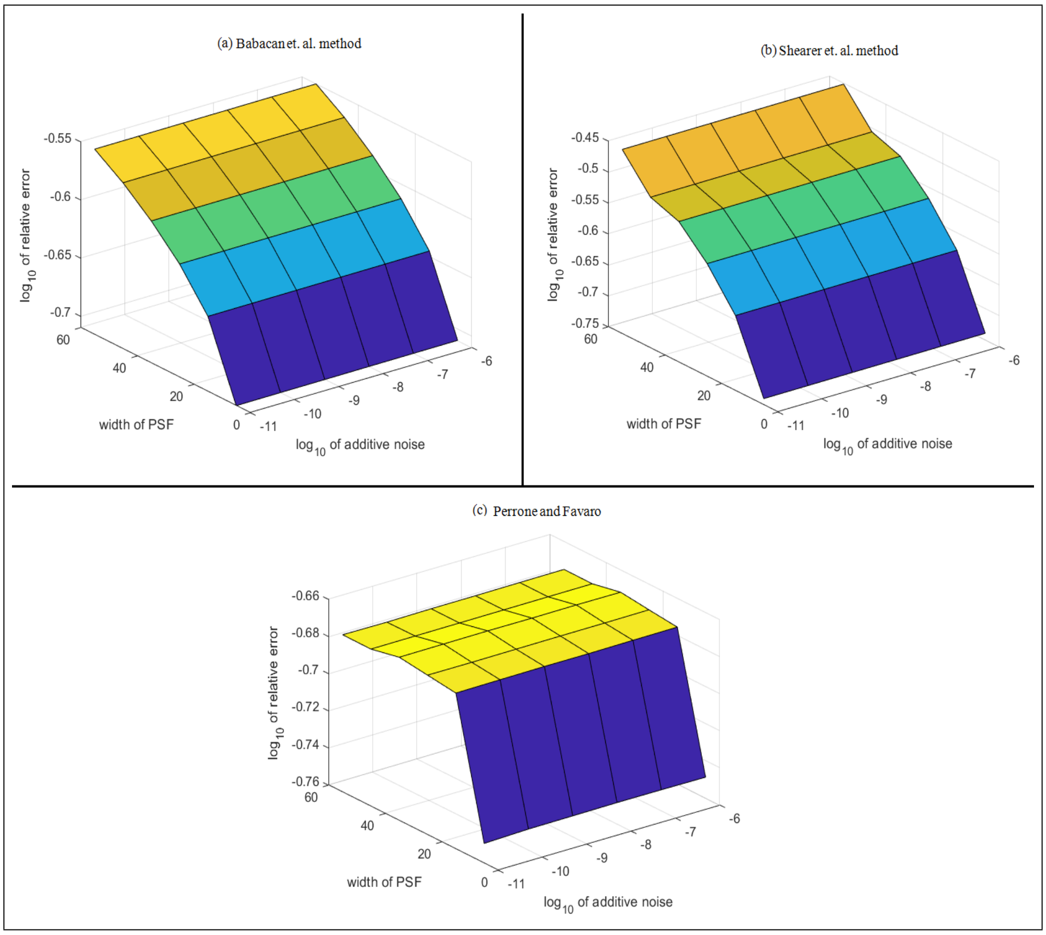


FIGURE 7.8: Bayesian statistical theory based methods error results of experiment one when a true image of grass has used

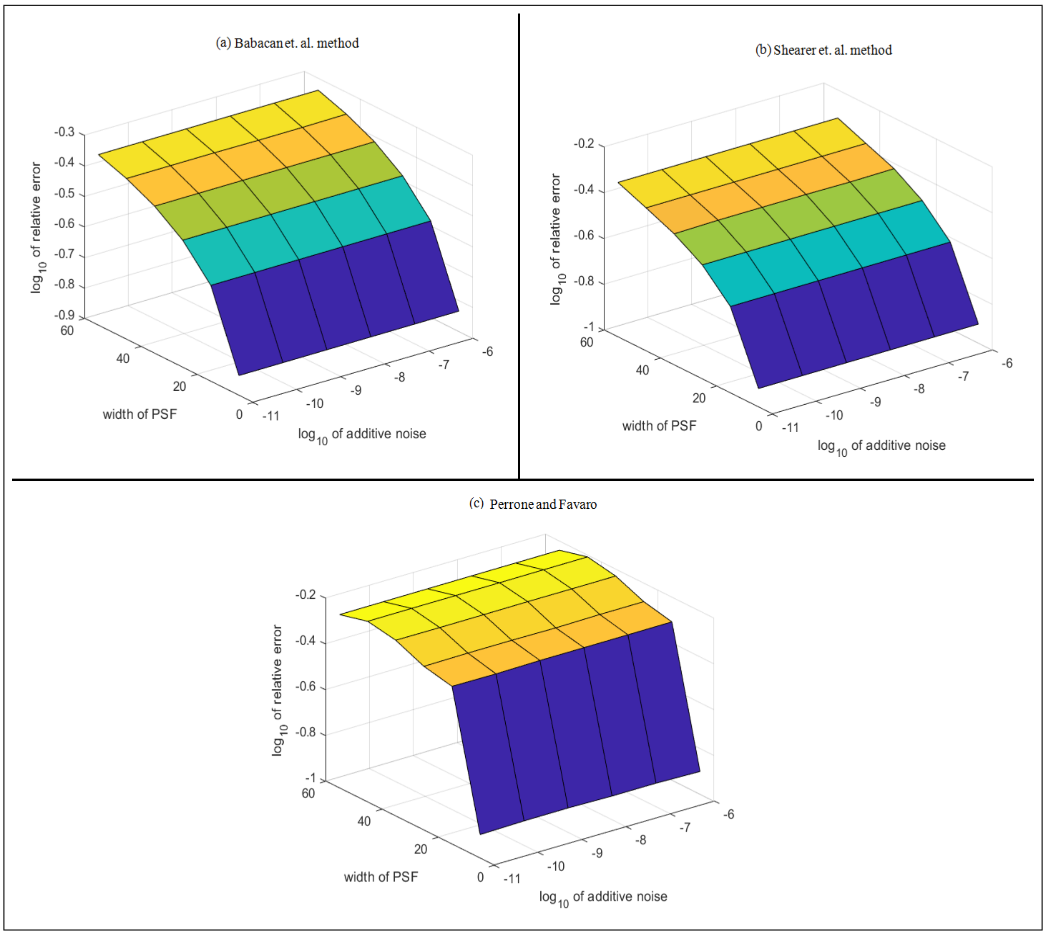


FIGURE 7.9: Bayesian statistical theory based methods error results of experiment one when a true image of girl face has used

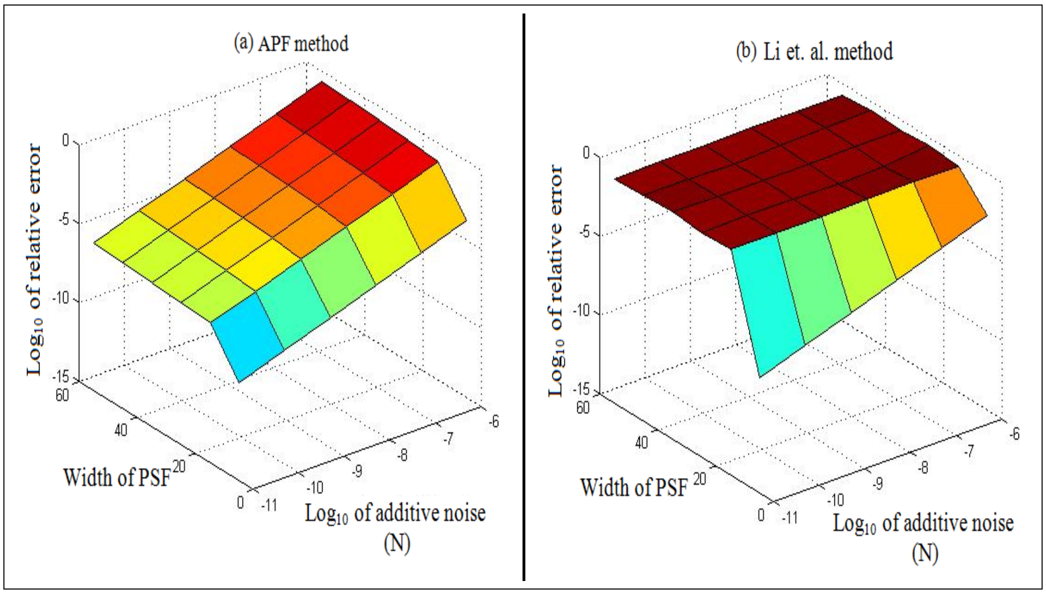


FIGURE 7.10: Linear algebra theory based methods error results of experiment one when a true image of San Diego has used

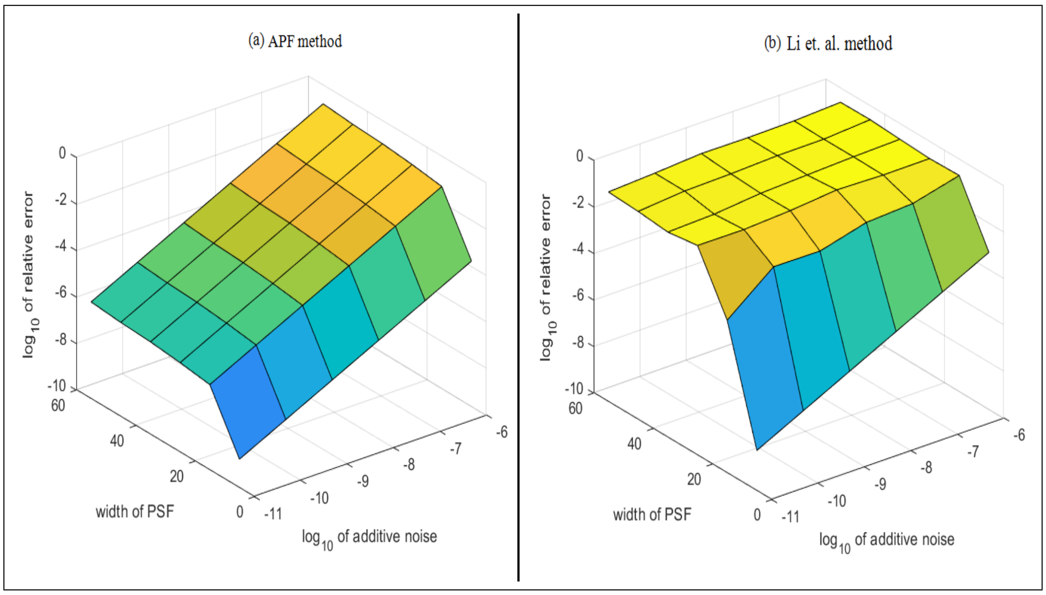


FIGURE 7.11: Linear algebra theory based methods error results of experiment one when a true image of cameraman has used

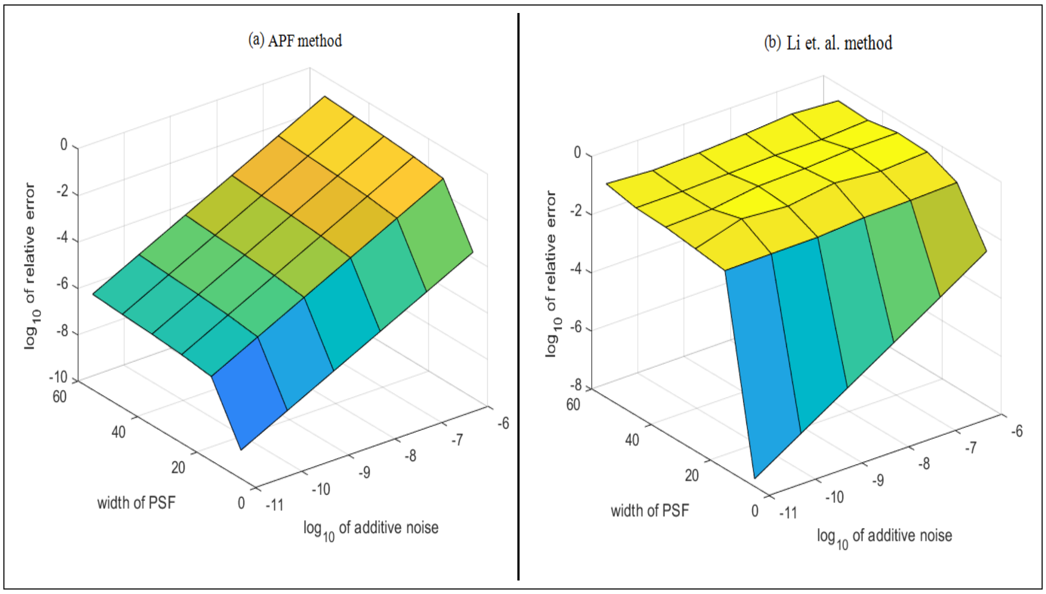


FIGURE 7.12: Linear algebra theory based methods error results of experiment one when a true image of girl face has used

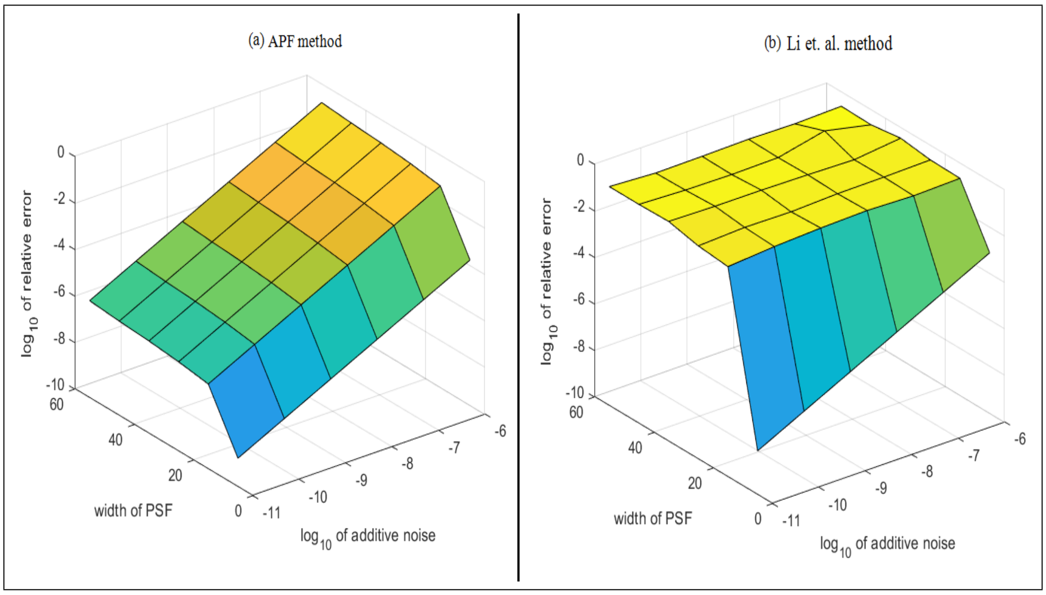


FIGURE 7.13: Linear algebra theory based methods error results of experiment one when a true image of grass has used

7.2.2 Experiment Two: PSF size error effects on BID methods' performance

The performance of four BID methods was investigated by varying the size of the PSF used in the algorithms. The same experiment was repeated on four different true images. The true images of Camera man, Lena, Pumpkins, and SanDiego's map are used. All true images are of size 180×180 pixels. The experiment started by processing each true image with a PSF of size 23×23 pixels, producing a set of four blurred images.

Each of the blurred images are then processed with four BID methods: Li et al., Shearer et al. [4], Babacan et al. [5] and Perrone and Favaro [3]. The size of the PSF used in the processing algorithms was then varied from 3 to 43, corresponding to an error in the range of -20 to $+20$ in the true PSF size used(23). The output of the processing algorithms and the recovered images are then compared to the true image and the relative error is computed. Figure 7.14 shows the results of the experiment for the four true images.

The four sub figures show that the relative error of the recovered images is considerably high for the largest negative PSF size error. As the size of the PSF is increased from 3 to 43 in the methods of Shearer et al. [4] and Babacan et al. [5], the relative error remained approximately constant throughout. However, in the method of Perrone and Favaro [3] the relative error performance worsened as the PSF size increased. The method of Li et al. [1, 2] showed similar behaviour to the method of Perrone and Favaro [3] except for when the size of the PSF used in the algorithm approached the true PSF size. The method of Li et al. [1, 2] gave a considerably better performance when the used PSF size in the algorithm matched the true PSF size.

These results show the importance of the accurate identification of the true PSF size. Inaccurate identification of the PSF size affects the performance of the method of Li et al. [1, 2] significantly. The theory of these methods claims that these methods are totally blind; however, it was found from the implemented algorithm that the PSF size is a parameter and is required in the image recovery process.

The performance of these methods suffers greatly from the lack of PSF size determination. In practice, the size of the PSF varies and is dependent on many variables. Assumptions about the size of the PSF become impractical and hence it is very beneficial to develop methods that identify the PSF size automatically as part of the image recovery algorithm. The developed method in this research study, the APF method, does exactly that. As Experiment One shows, this method is able to identify correctly the size of the PSF from a small size up to a very large size of 55 × 55.

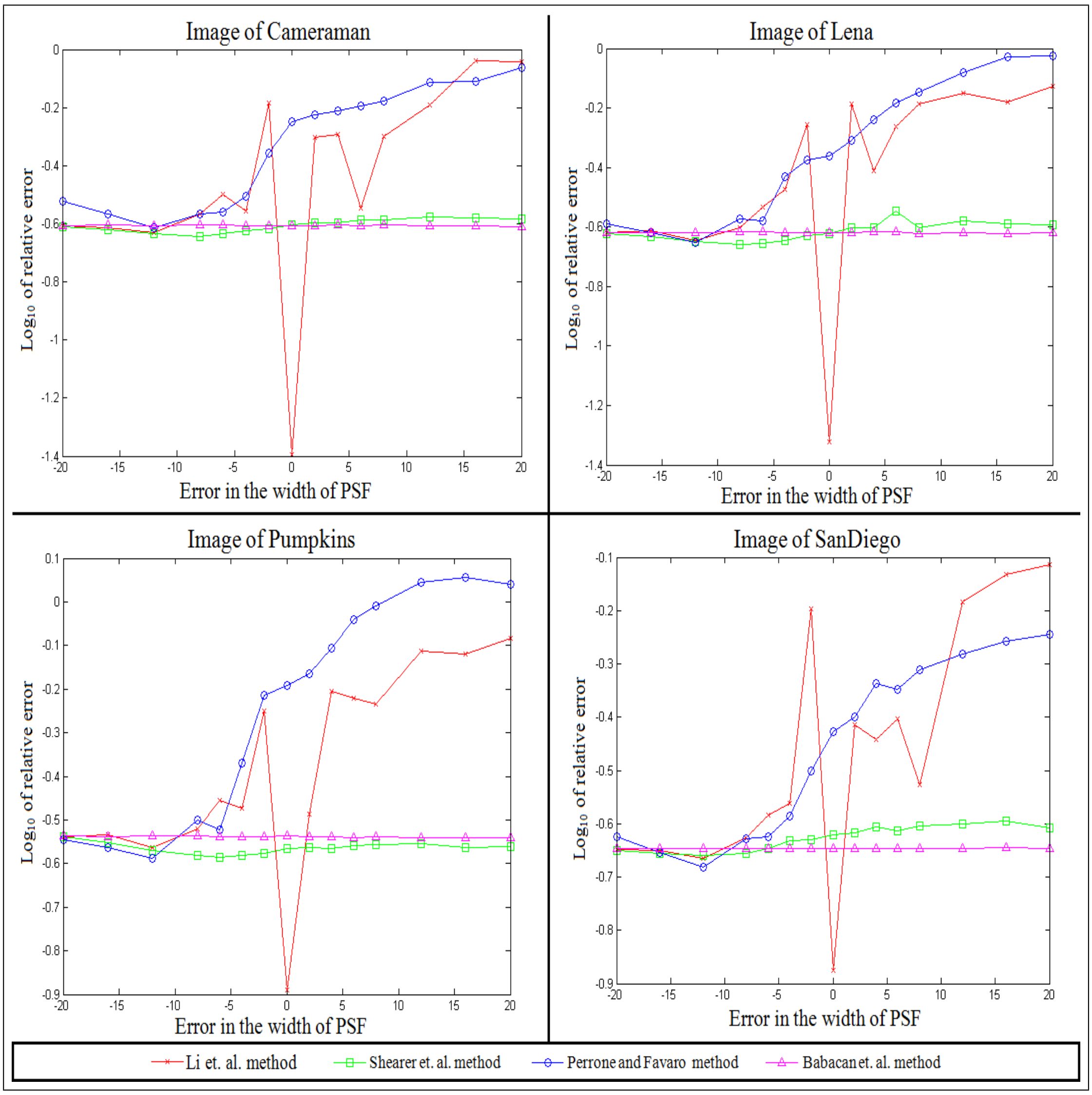


FIGURE 7.14: BID methods relative error performance against used PSF size error

The method of Li et al. [1, 2] showed sensitivity to the error in the PSF size; in particular, the performance of the method improved when the error in the PSF size decreased to zero. This is to be expected as algebraic-based methods systematically identify the PSF from the given data. In contrast, statistically-based methods require a prior probability distribution and hence the results of the recovered images depend on this prior because different prior yield different deblurred images. This would result in a significant number of failed recovered images, due to its high sensitivity to the prior value, which demands high effort and in some cases is not possible to compute accurately.

7.2.3 Experiment Three: algebraic methods PSF determination performance

The recovery of the real images from the blurred images is affected significantly by the accuracy of the computed PSF. In other words, the correct identification of the PSF is an important factor in the recovery process of the true images. In this experiment, the performance of methods of Li et al. [1, 2] and the APF in recovering the true PSF is considered. A true image of a girl's face of size 180 × 180 is processed with two known PSFs of sizes 15 × 15 and 25 × 25. In addition, the following two sets of noise levels and PSF additive measurement errors are used during the blurred image generation:

Additive noise levels: $N = 0, 10^{-11}, 10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}$. PSF measurement error levels: $E = 0, 10^{-11}, 10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}$.

Forty-nine blurred images are then produced for each PSF using the following processing algorithm:

$$\mathcal{G} = (\mathcal{H} + \mathcal{E}) \otimes \mathcal{F} + \mathcal{N}, \tag{7.2}$$

Each one of the ninety-eight blurred images, corresponding to the two true PSFs, is then processed through the methods of Li et al. [1, 2] and the APF. The computed PSFs are then compared to the true PSF using a relative error criterion for each method discussed in (2.39). Figure 7.15 shows the relative error results for both the methods of Li et al. [1, 2] and the APF.

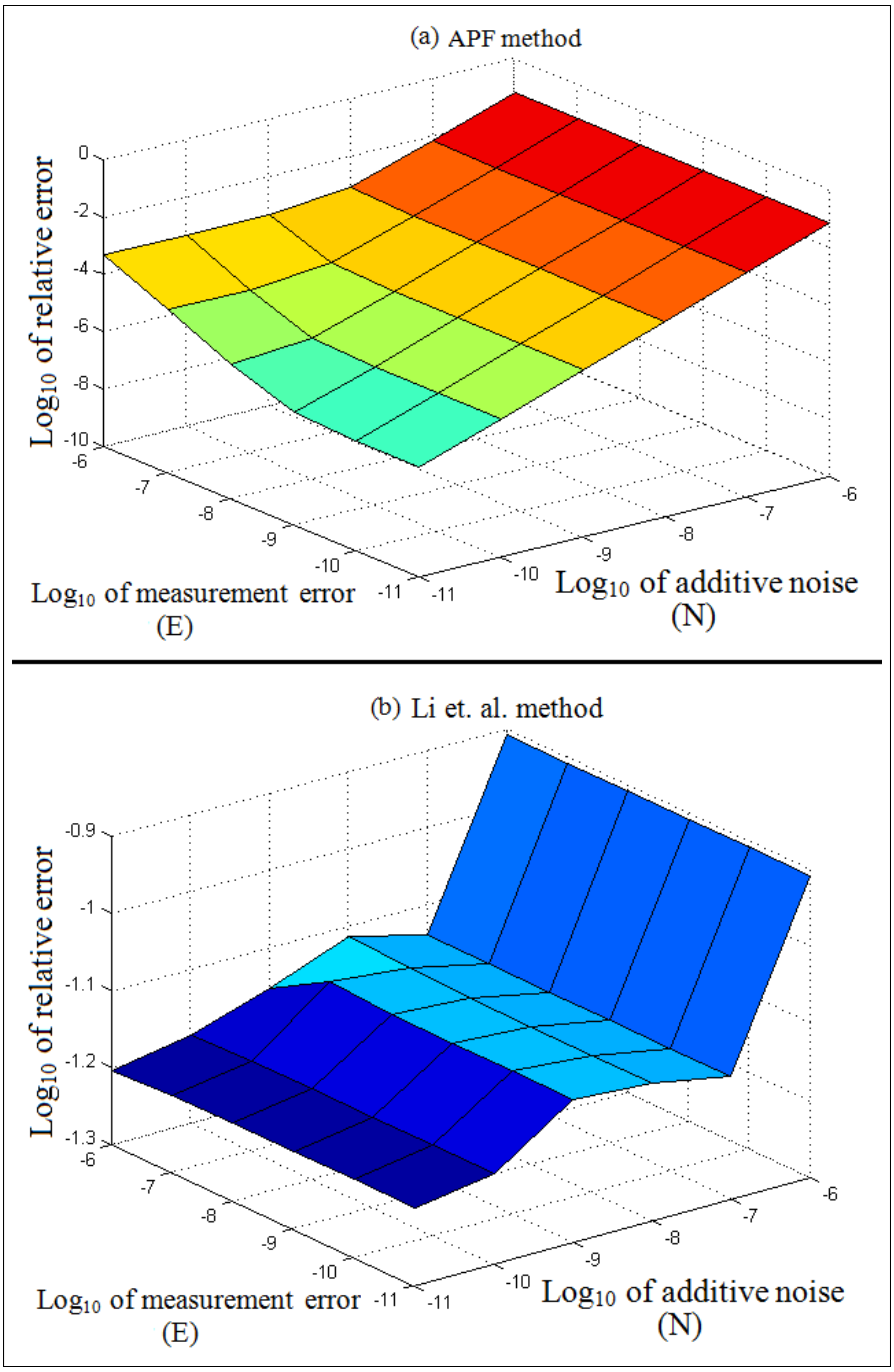


FIGURE 7.15: Relative error comparison between the true PSF and the computed PSFs using the methods of Li et al. [1, 2] and the APF

The figure shows that the relative error of the method of Li et al. [1, 2] is consistently higher than that in the APF method. In particular, the highest relative error reported by the APF method is of the same order as the lowest relative error reported by the method of Li et al. [1, 2]. In other words, the worst performance, the case with high levels of additive and measurement noise, for the APF is similar to the best performance of the method of Li et al. [1, 2] in the case of no additive and measurement noise.

From the figure, it can be observed that the relative error patterns for the two PSFs are approximately the same for the APF method. In contrast to this, for the method of Li et al. [1, 2], the pattern of the relative error is inconsistent and depends on the true PSF. This shows that the computation of the PSF, in the APF method, is not affected by the size of the true PSF.

7.2.4 Experiment Four: Feature detection performance

Feature detection is an important area of image processing. Images are normally pre-processed to remove blur from them before feature detection processing commences. The performance of the detection algorithms depends on the quality of the pre-processed (deblurred) images. In this section the performance of the deblurring algorithms are examined using feature detection methods. Face feature detection and circle feature detection experiments are performed on deblurred images using various deblurring algorithms. In each section, the quality of the feature detection, and hence the quality of the deblurring algorithm, is compared to many of the state-of-the-art deblurring algorithms.

7.2.4.1 Face feature detection performance

A true image of a group of eight people is used in this experiment. The image is of size 550 × 200 and shows eight faces clearly visible. This image is then blurred with a Gaussian blurring function of size 25 × 25 (see Figure 7.16). A cascade object detector function based on the Viola-Jones algorithm is used. This algorithm is implemented in the computer vision system toolbox in MATLAB. The true image was first processed through the feature detector and the results are shown in the top part of Figure 7.16. The figure shows that the detector successfully identified all the eight faces in the true image as expected. The bottom part of the figure shows the feature detector results when the blurred image is used instead of the true image. The results show that only one face was detected successfully. This establishes a baseline for the feature detector performance against which the performance of the deblurring algorithms can be based.

Nine deblurring algorithms are then used to recover the true image from the blurred image (bottom of Figure 7.16) before processing it through the feature detector. Figure 7.17 shows the detection performance on images deblurred using the basic deblurring algorithms. The detection performance when the Richardson-Lucy and ML deblurring algorithms are used is the same, in which the same three faces were identified. Using the Regularisation algorithm to deblur the image before processing it through the feature detector improved the detection considerably. Six out of the eight faces were detected in this case. The best detection performance, using the basic methods, was achieved using the Wiener algorithm where seven faces were detected.

The performance of the feature detector on a deblurred image using the statistical methods, methods of Shearer et al. [4], Babacan et al. [5] and Perrone and Favaro [3] are shown in Figure 7.18. The results show that the use of the Perrone and Favaro algorithm to deblur the image worsened the feature detector's performance while the use of the Babacan et al. [5] algorithm made no difference to the feature detection. The exception of these methods is Shearer et al. [4]. Using the deblurring algorithm of Shearer et al. [4], the performance of the detector improved with a detection of four faces out of the eight faces. The performance of the statistical deblurring methods in general was worse than that of the basic deblurring algorithms. However the performance of the Shearer et al. [4] deblurring algorithm was marginally better than the Richardson-Lucy and ML methods, with four faces being detected rather than three.

Figure 7.19 shows the results obtained from the feature detector when the algebraic deblurring methods were used to recover the image. The feature detector identified

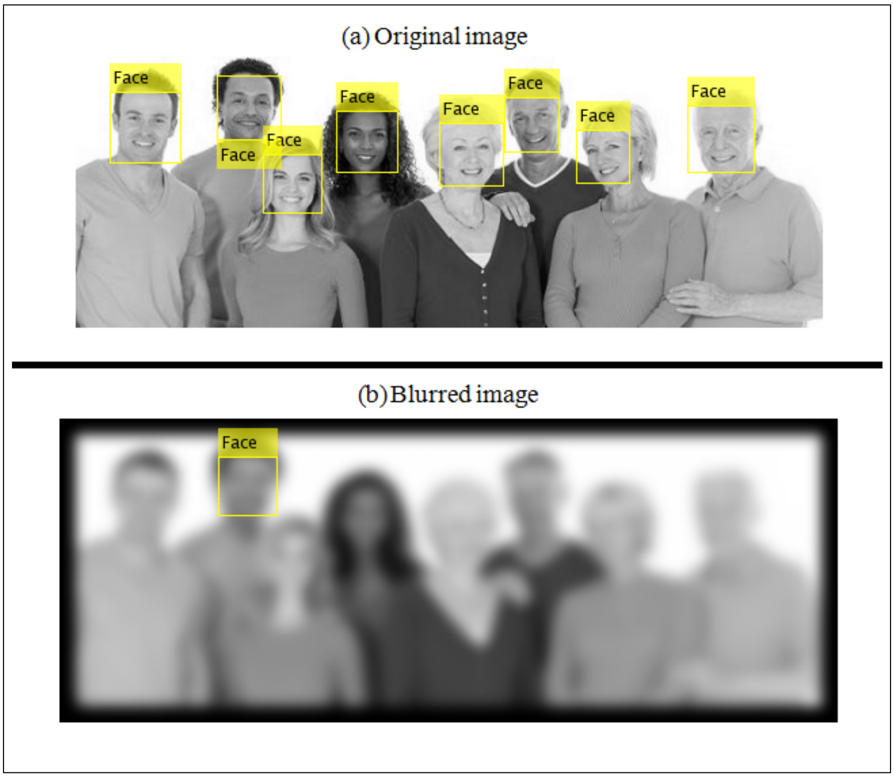
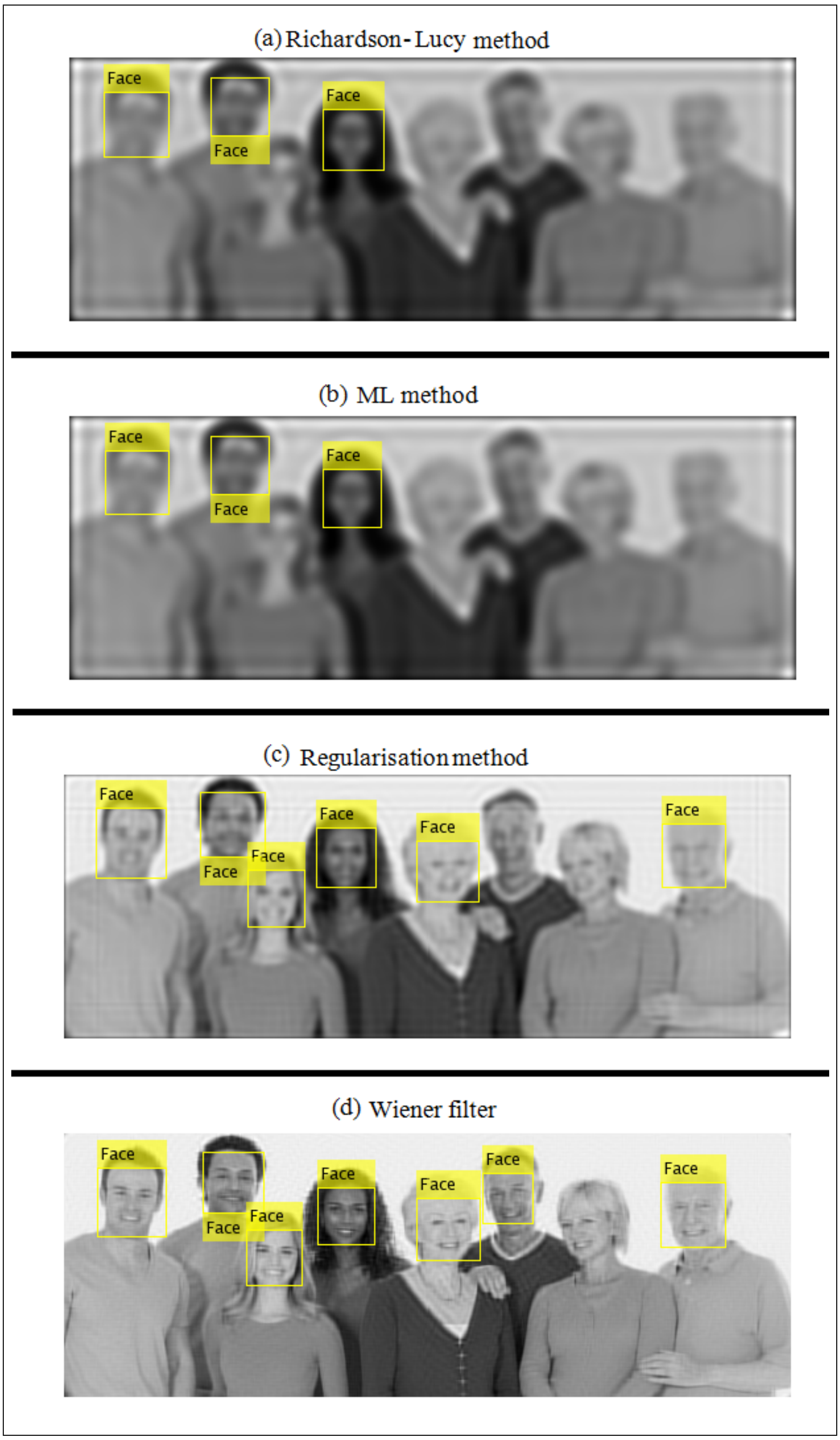


FIGURE 7.16: The face detector results using (a) the original image and (b) the blurred image.

all eight faces in the deblurred image. This shows that the algebraic methods are far superior to the statistical and the basic state-of-the-art deblurring methods.

In order to further confirm the performance of the deblurring algorithms, the true image was processed with an increasing size of PSF function, from size 3 × 3 to size 55 × 55. Each of the blurred images is then recovered using the nine deblurring algorithms. The recovered images are then processed with the face feature detector. The ratio of the number of faces detected are plotted in Figure 7.20.

The figure shows that when the statistical methods are used to recover the true image, the face detection performance deteriorated as the PSF increased. The detection ratio dropped to zero faces as the PSF size increased to size 23 × 23. The performance of the Richardson-Lucy algorithm and the ML algorithm is identical. The face detection ratio for these methods started with a ratio lower than one (all faces detected) and remained constant until a PSF size of 23 × 23 where the detection ratio dropped to zero and remained zero for higher PSF sizes. The regularization and Wiener algorithms showed


(a) Richardson-Lucy method

(b) ML method

(c) Regularisation method

(d) Wiener filter

FIGURE 7.17: The face detector results on a deblurred image using the BID methods with their function implemented in MATLAB.

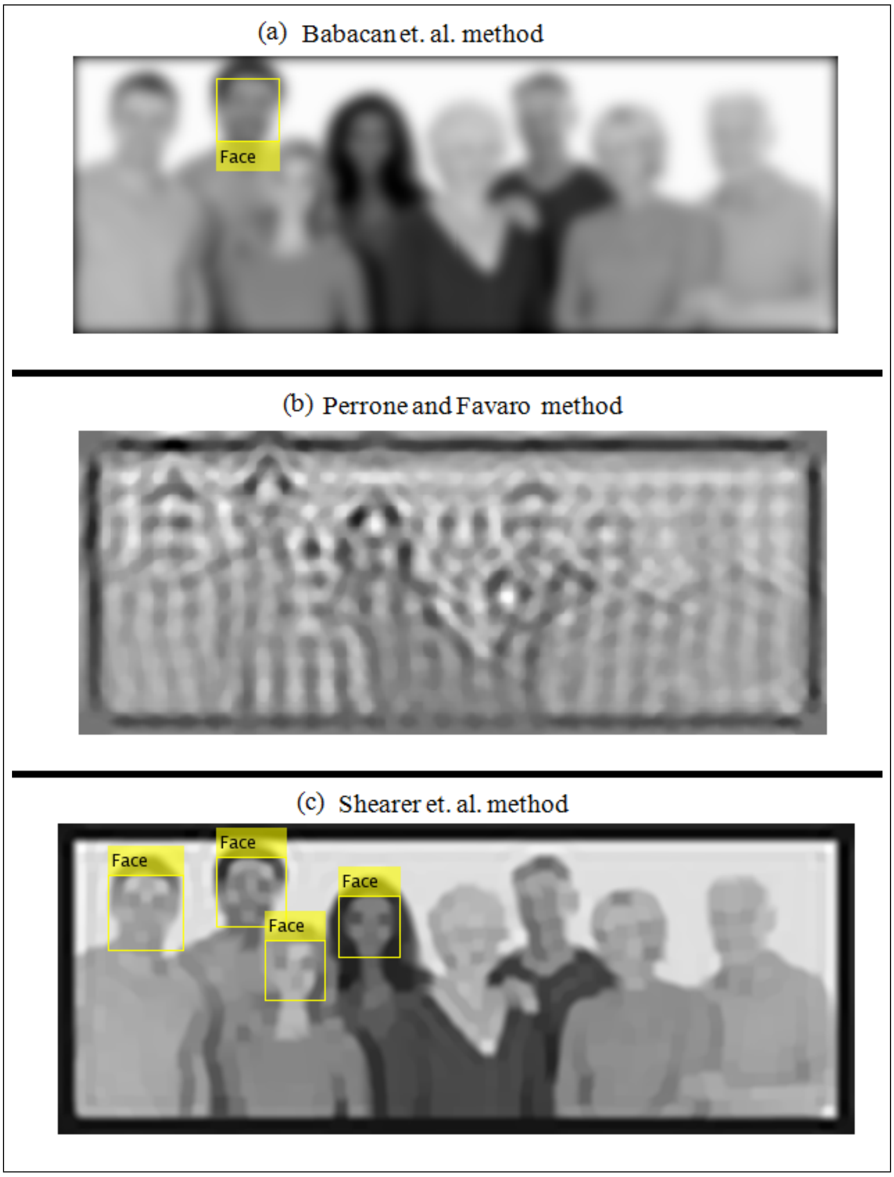

(a) Babacan et. al. method
Face
(b) Perrone and Favaro method
(c) Shearer et. al. method
Face
Face
Face
Face
Face

FIGURE 7.18: The face detector results on a deblurred image using the BID methods based on the statistical methods.

better performance than the other basic and statistical algorithms. The detection ratio of the Wiener algorithm remained above 50% with a performance better than 50% for smaller PSF sizes. The regularization algorithm showed an identical performance to the Wiener algorithm for PSF sizes less than or equal to 23 × 23. However, the ratio of detection started to deteriorate as the PSF size increased and eventually the ratio dropped to zero at PSF size 43 × 43.

The performance of the linear algebra-based methods is the best, by far, among all the compared deblurring algorithms, as shown in the figure. The detection rate remained

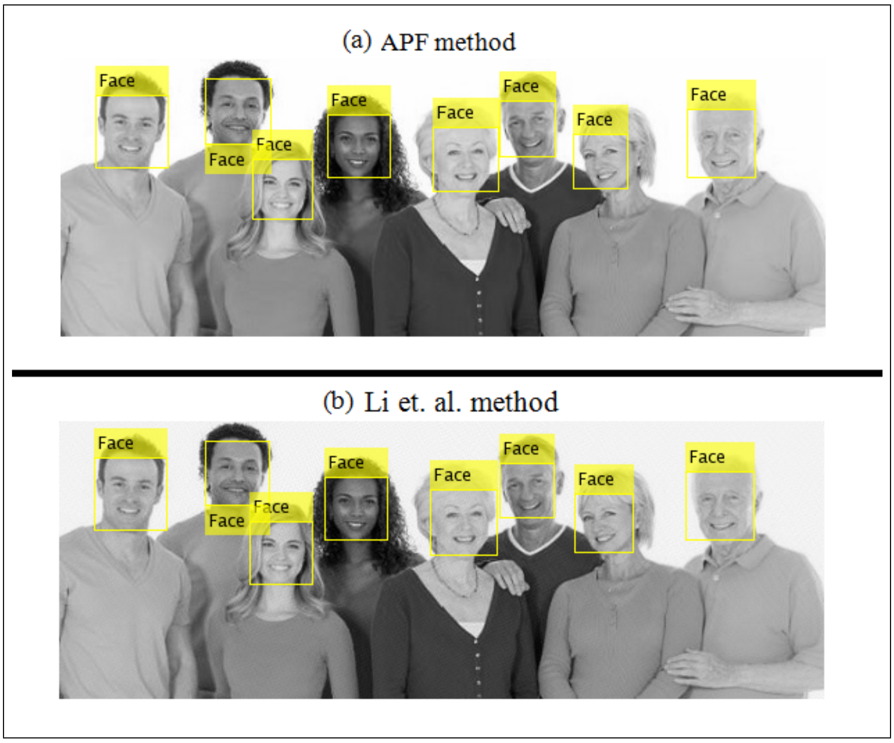


Figure 7.19: The face detector results on a deblurred image using the BID methods based on linear algebra methods.

close to 100%. The APF algorithm showed 100% performance for all the PSF function sizes. The algorithm performance of Li et al.'s [1, 2] method fluctuated between 90% and 100% detection rates. This shows that the developed APF algorithm image recovery quality is very close to the true image. The face detection ratio remained constant with a 100% detection rate for all PSF sizes, from size 3 × 3 to size 55 × 55. Real blurred images are rarely blurred with PSF of size 55 × 55 or more. This shows that the APF method is faithfully able to deconvolve the components of the blurring function from the components of the true image even when the blurring is high.

7.2.4.2 Circle feature detection performance

In this section the effects of additive noise $\mathcal{N}$ (error in the image) and measurement error $\mathcal{E}$ (error in the true PSF) on the deblurring algorithms' performance is investigated through the use of a circle detection algorithm. The noise and measurement error in the blurred image are modelled as follows:

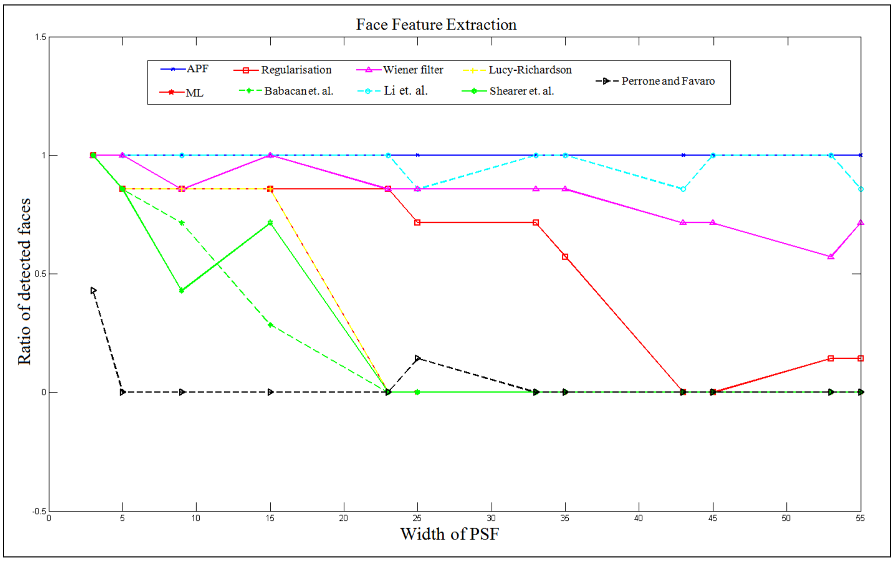


FIGURE 7.20: The performance of the face feature detector on a deblurred image using BID methods.

$$\mathcal{G} = \mathcal{F} \otimes (\mathcal{H} + \mathcal{E}) + \mathcal{N}. \tag{7.3}$$

The circle detection algorithm used is the *imfindcircles.m* function implemented in MATLAB image processing toolbox. This function uses the circular Hough transform to find the centres and radii of circles in an image. This function takes a range parameter that specifies the minimum and maximum value for the radii of interest.

For this experiment a true image of size 300×205 is used. The image contains a number of circles with different radii and centres as shown in Figure 7.21. This image is processed through the circle finder, which returned a set of eighteen circles. For each circle the center and radius is returned in float number format.

To measure the performance of the deblurring algorithms, the true image is processed with a separable Gaussian PSF of size 35×35, measurement error of $\varepsilon = 1 \times 10^{-8}$, and an additive noise of $\varepsilon = 1 \times 10^{-7}$. The resulting image is then used as an input for the nine deblurring algorithms.

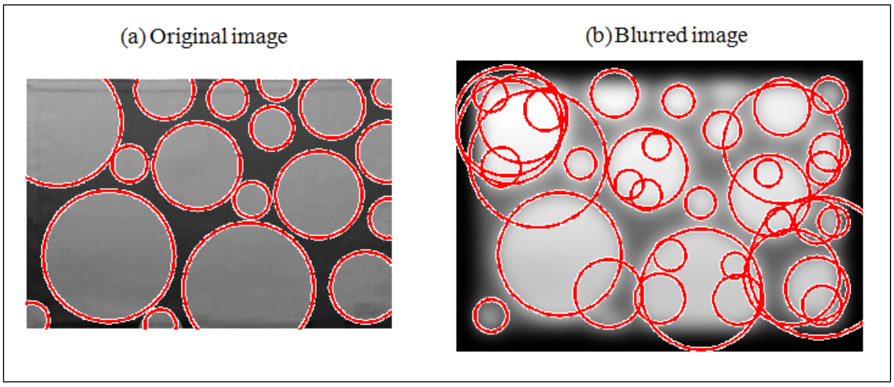


Figure 7.21: The result of the circle detector using (a) the original image and (b) the blurred image.

The deblurred images that result from the nine deblurring algorithms are then processed in the circle finder algorithm. The detection performance when the Richardson-Lucy, Regularisation, ML and Wiener deblurring algorithms are used is shown in Figures 7.22. Sixty-five out of the eighteen circles were detected when the Richardson-Lucy and ML algorithms were used. Using the Regularisation algorithm to deblur the image before processing it through the feature detector resulted in worsening the detection. Seventy-six circles were detected in this case. The best detection performance, using the basic methods, was achieved using the Wiener algorithm, where thirty-eight circles were detected.

The performance of the feature detector on a deblurred image using the statistical methods, methods of Shearer et al. [4], Babacan et al. [5] and Perrone and Favaro [3] are shown in Figure 7.23. The results show that the use of the method of Shearer et al. to deblur the image worsened the feature detector performance by detecting seventy-eight circles while the use of Babacan et al.'s [5] algorithm provided forty-three detector circles. The use of Perrone and Favaro's method to deblur the image provided sixty-two detector circles.

The performance of the linear algebra-based methods is the best, by far, among all the compared deblurring algorithms, as shown in Figure 7.24. The APF algorithm was able to detect the eighteen circles with no extra circles being detected, while Li et al.'s [1, 2] method detected nineteen out of the eighteen circles. This shows that the developed

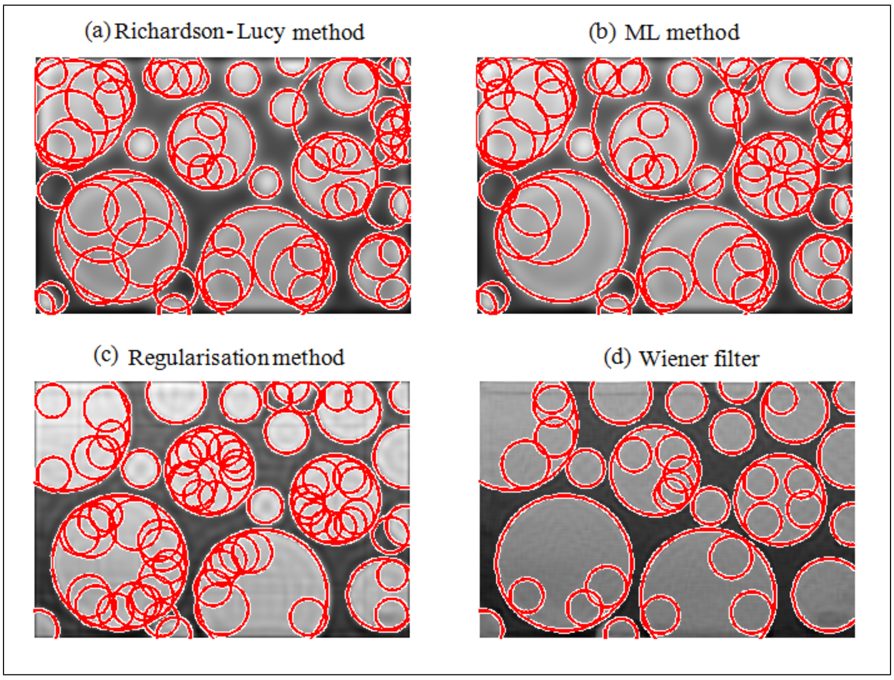


FIGURE 7.22: The result of the circle detector on a deblurred image using the BID methods with their function implemented in MATLAB

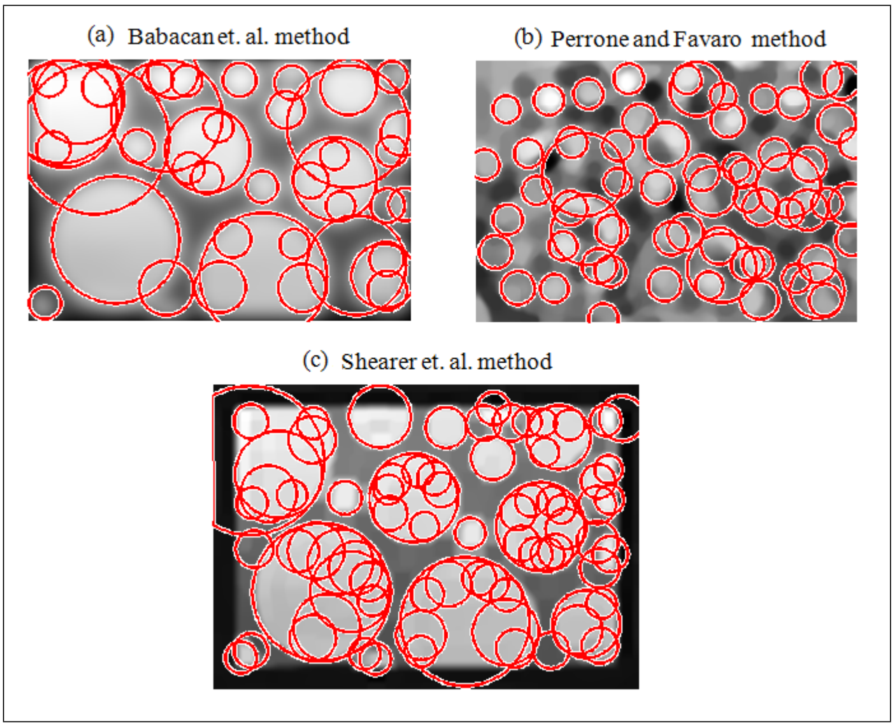

(a) Babacan et. al. method
(b) Perrone and Favaro method
(c) Shearer et. al. method

FIGURE 7.23: The results of the circle detector on a deblurred image using the BID methods based on the statistical methods.

APF algorithm image recovery quality is very close to the true image. This shows that the APF method is faithfully able to deconvolve the components of the blurring function from the components of the true image even in the presence of high additive noise and measurement error.

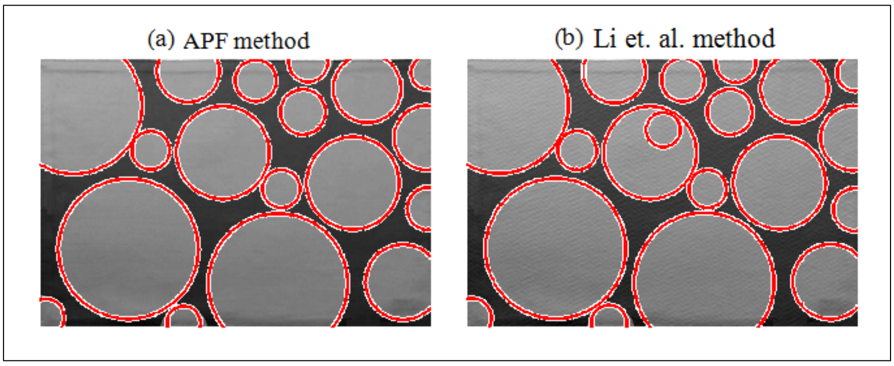


FIGURE 7.24: The circle detector results on a deblurred image using the BID methods based on linear algebra methods.

The results of the circle finder are then compared and plotted in Figure 7.25. The horizontal axis represents the deblurring method used whereas the vertical axis shows the number of circles found. Three colours are used to categorize the results of each deblurring algorithm. A blue colour represents the number of circles detected exactly, and a green colour represents the circles found within a tolerance of ±0.5 from the true value. Circles that are detected outside the tolerance range are represented by a red colour.

From Figure 7.25, the results of the APF method are shown in the form of thirteen found circles in blue. This means that the thirteen found circles in the deblurred image are exactly the ones found in the true image. The figure shows clearly that all the other deblurring algorithms resulted in a performance that is sub-optimal from that of the APF algorithm.

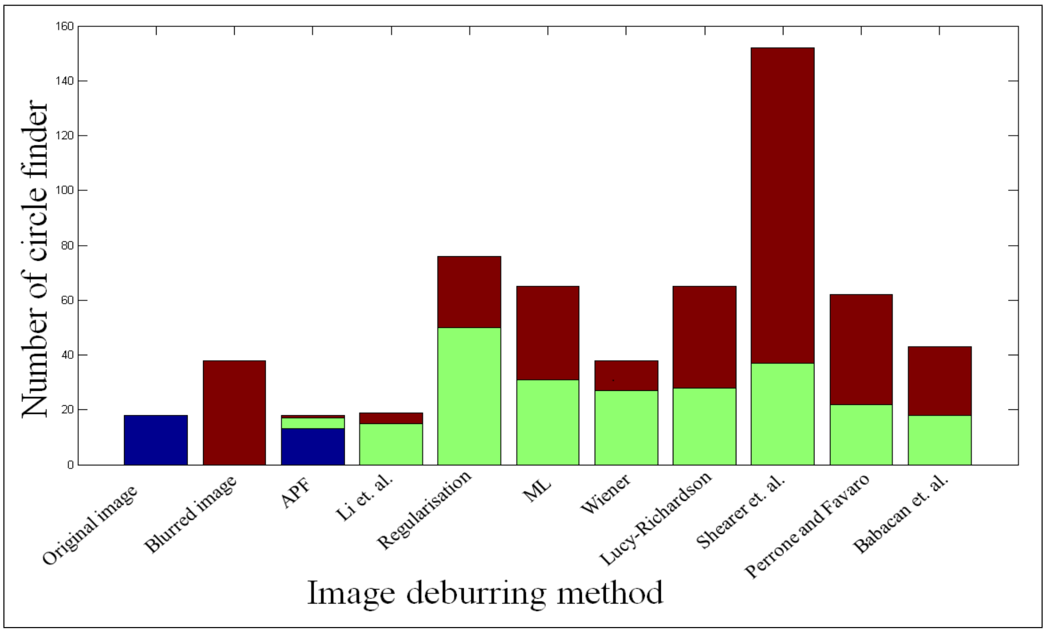


Figure 7.25: The performance of the circle feature detector on a deblurred image using BID methods.

7.3 Real Images Processing

In this section the application of the developed algebraic based BID method to three real blurred images without pre-processing is considered. The three real images used in this experiment are extracted from Anat Levin database [56, 59]. These images (image1,image2 and image3)are of size 255 × 255 and shown in Figure 7.26.

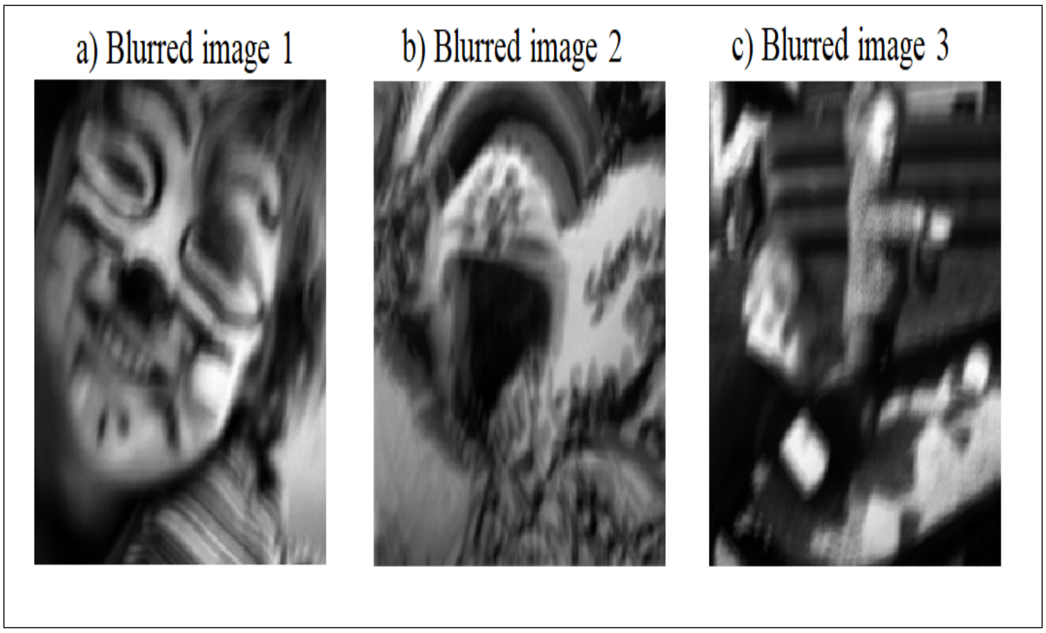


FIGURE 7.26: Real blurred image.

The developed BID method was applied to the above three images using the developed AGCD algorithm. The results of the application of the developed method in the spatial domain on the three real images failed. The algorithm failed with an exception failure due to NaN in the AGCD coefficient computation. Once this exception occurred the algorithm stops and no more results could be obtained. This exception occurred on all the three experimented images.

The frequency domain application of the AGCD algorithm on the above three images is considered next. In this method the images are loaded and transformed into the frequency domain before the application of the AGCD algorithm. In this case no exception failure was encountered, however the resulted deblurred images are visually worse than the blurred real images. The results of the frequency domain deblurring of the three images is shown on Figure 7.27.

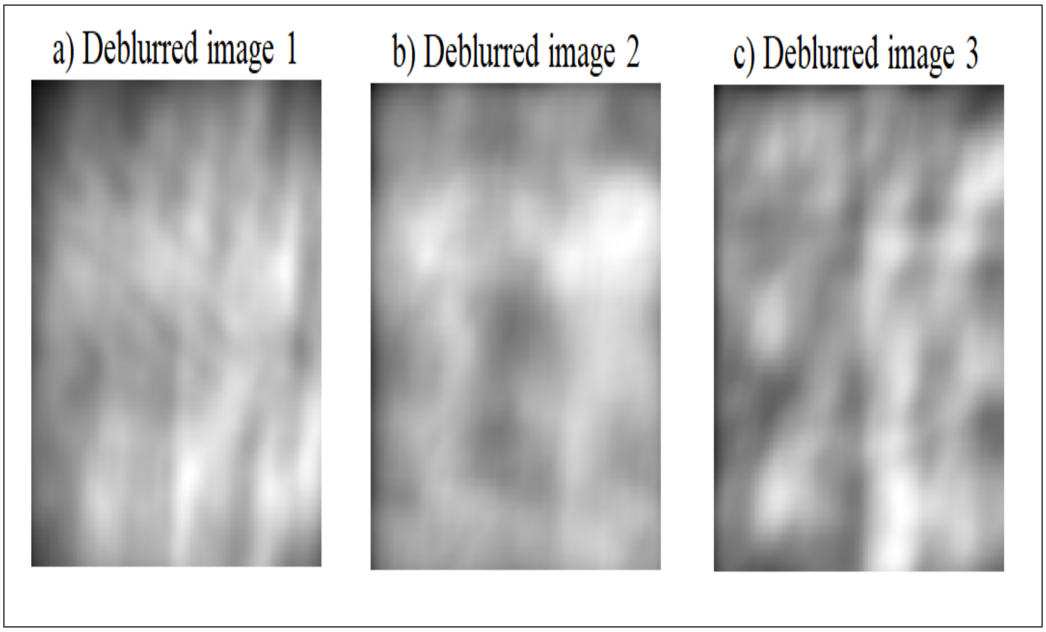


FIGURE 7.27: The result of deblurring the real blurred image.

The above results could have been expected as the algebraic methods are more sensitive to errors than statistical methods in general. In addition, the developed AGCD algebraic method requires the presence of the image boundary pixels which results due to the process of blurring. These boundaries size depends on the type of convolution and the size of the PSF, as mentioned on Chapter 1. Therefore, a first step for a successful AGCD deblurring of real blurred images, the boundaries need to be computed first before applying the AGCD method.

7.4 Summary

The developed BID method is designed to be robust and computationally efficient in the estimation of the PSF. This is achieved by implementing an AGCD computation method using APF, which is considered in Chapters 4 and 5. This chapter has examined and compared the performance of the developed APF method against the state-of-the-art methods. The analysis and the three sets of experiments considered in this chapter show the superiority of the developed method over the state-of-the-art methods in deblurring artificially blurred images. The experiments in this chapter examine and compare the performance of the BID methods in recovering the original image by considering different levels of additive noise, measurement errors and the size of the PSF. Four experiments have been carried out, using Matlab. Experiment one examines the performance of the BID methods, including the developed method, in relation to changes in PSF width and additive noise. Experiment two examines the behaviour of the state-of-the-art methods in recovering the true image from a blurred image with an unknown PSF size. The third experiment examines the performance of the developed method against other state-of-the-art linear algebra methods. Images with different additive noise levels and errors in PSF were used in this experiment. In all three experiments, an image size of 180 × 180 is used.

The application of the developed AGCD BID method to three real blurred images failed consistently on all three images. In the spatial domain the developed algorithm failed with NaN exception failure while in the frequency domain the developed method failed in resulting in a deblurred image that is visually worse than the input image. This points to the sensitivity of the developed method to boundary condition and the need to develop a robust algorithm for the computation of these boundaries before applying the developed method.

Experiment four examines the performance of the deblurring algorithms for feature detection. Face feature detection and circle feature detection experiments are performed on deblurred images from various deblurring algorithms. Using relative error as a criterion, the error level of the APF method in recovering the true images is considerably lower than the other developed methods. Also, the APF method show consistency in computing the PSF and in particular the error in the computation does not depend on the true size of the PSF.

Chapter 8

Conclusions and Future Work

8.1 Introduction

The blind image deconvolution problem is the central subject of this doctoral thesis. There are a number of approaches for solving the BID problem, including basic methods, that were implemented in the image processing toolbox in Matlab, statistical methods, which are recently developed, and linear algebraic methods. The approach adopted in this research study for solving this problem falls within the class of linear algebraic methods. As the acronym of BID suggests, there should be no prior knowledge assumed about the true image and the PSF. The algebraic approach offers a way of solving the BID problem without requiring any prior knowledge about the true image. This chapter presents the overall conclusions for the adopted approach in this research study for solving the BID problem and also provides suggestions for future research.

8.2 Conclusions

Images play a significant and important role in diverse areas of everyday modern life. Examples of the areas where the use of images is routine includes medicine, forensic investigation, engineering applications and astronomical science. The procedures and methods that depend on image processing would benefit considerably from blur-free images. Most of the images, unfortunately, are affected by noise and blur that result from the practical limitations of image sourcing systems. The blurring and noise effects render the image less useful. An efficient method for image restoration is hence important for many applications.

The restoration of true images from blurred images is the inverse of the naturally occurring problem of true image convolution by a blurring function. The deconvolution of images from blurred images is a non-trivial task. One challenge is that the computation of the PSF is an ill-posed problem, i.e. an infinite number of solutions are possible. Another challenge is that deconvolution is ill-conditioned. Small perturbations in the blurred image can lead to a large deviation from the true image. A further challenge is that most of the practical applications of image restoration require computationally fast and accurate algorithms in order to be of any practical use.

The performance of the state-of-the-art image restoration methods varies in terms of computational speed and accuracy. In addition, most of these methods require prior knowledge about the true image and the blurring function which is, in a significant number of applications, an impractical requirement. The development of image restoration methods that require no prior knowledge about the true image and the blurring functions is hence desirable. Previous attempts of developing BID methods resulted in methods that have robust performance against noise perturbations. Their good performance is, however, limited to PSFs of small size. In addition, even for PSF of a small size, these methods required the size of the PSFs to be known and an estimate of the noise level present in the blurred image.

A BID method that truly requires no knowledge of the PSF size and noise level present in the blurred image as well as offers good accuracy and a speedy performance is highly desirable. Polynomial linear algebra offers a way of computing the PSF size and its components without requiring any prior knowledge about the true image and the blurring PSF. This research study has developed a BID method for image restoration based on

AGCD computation, specifically, the approximate polynomial factorization APF algorithm. The developed method used the structure preserving Sylvester resultant matrix algorithm in the computation of the AGCD and QR decomposition for computing the degree of the AGCD.

Representing images as bivariate polynomials, and assuming that the blurred image is the result of the convolution of a PSF bivariate polynomial and the true image bivariate polynomial, enables the use of polynomial linear algebra techniques to extract the true images from the blurred images. The AGCD represent the PSF and the true image is the polynomial division of the blurred image by the PSF.

In practice, PSF can be spatially variant or invariant. PSF spatial invariance means that the blurred image pixels are the convolution of the true image pixels and the same PSF. Some of the PSF bivariate functions, separable functions, can be further simplified as the multiplication of two univariate polynomials. This research study is focused on the invariant separable and non-separable PSF cases.

The developed BID method for restoring images consists of three main processing steps. The first step is the AGCD degree calculation, followed by the coefficient calculation of the AGCD, and finally the deconvolution of the true image from the blurred image. Experiments have been designed and performed to measure the performance of the developed method and the results were compared to the state-of-the-art methods. The results show a significant improvement in calculating the PSF size and its component values as well as an improvement in the restored images quality under perfect boundary condition. This method allows for the computation of the PSF of variable sizes and is not limited to small sizes.

Further research work and suggestions are considered in the next section.

8.3 Suggestion for Future Research

As stated in the previous section, this research study has focused on the case of spatially invariant PSF. An important case in image deblurring is that of the spatially variant PSF. The application of the algebraic method for the spatially variant PSF was not considered in this research study. A future development of the spatially variant PSF case in the context of algebraic method is needed to further evaluate the applicability of algebraic method to the BID problem. This area of research is challenging and requires the development of methods that can compute the PSF size and its component values in different regions from the blurred image.

Two dimensional convolution, under perfect boundary conditions, has been used to examine the feasibility of the structure preserving Sylvester resultant matrix and the Approximate Polynomial Factorisation (APF) for totally BID problem. Natural images processing requires the estimation of the image's boundaries. This is another challenging area for further research. A proposed framework to tackle this area of research is shown in Figure 8.1. Future research in this area should investigate the feasibility of the existing extrapolation method in restoring blurred image boundaries or develop methods that are capable of estimating these boundaries. A boundary size estimation method needs to be developed in order to extrapolate the boundary pixel value. The performance of the developed method in this research study should be investigated in relation to this case of non perfect or estimated boundary conditions.

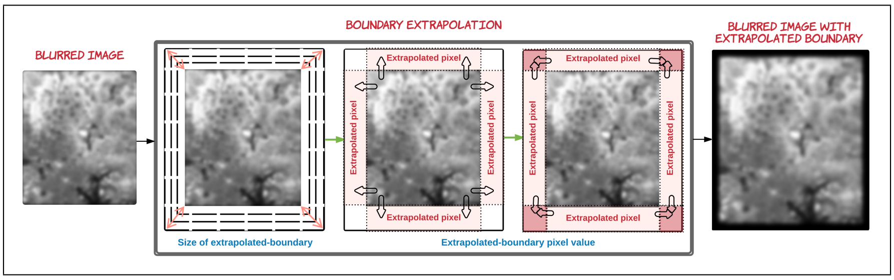


FIGURE 8.1: Proposed boundary extrapolation framework for naturally blurred images.

Further research studies should consider the performance of the algebraic based BID methods in the presence of various levels of additive noise. The research should consider the existing denoising methods and their applicability in the deblurring of images using algebraic methods and the developed method in this research study in particular. It would be beneficial to see how these methods perform in the presence of a moderate level of noise. This could be done by denoising naturally or artificially blurred images before processing them by the developed BID method in a pre-processing stage.

8.4 Summary

In summary, blind image deconvolution is a non-trivial task. The representation of images through bivariate polynomials opened the way for using linear algebra techniques to solve this problem, under perfect boundary conditions. This research study implements an efficient algorithm for BID problem, such that prior knowledge of the blurring function (PSF) and exact image are not required. The proposed algorithm is designed to incorporate the identification of PSF and the restoration of the image in two separate computational algorithms. The first algorithm computes the blur function from a blurred image, and is then used in the second algorithm to remove the blur from the blurred image. This is done by representing the blurred and deblurred images, and the PSF as bivariate polynomials in which the coefficients are the pixel values. The PSF is formulated as the approximate greatest common divisor (AGCD) of two degraded images that are represented by two bivariate polynomials. This research is carried out using approximate polynomial factorisation (APF) computational algorithms using a structure-preserving Sylvester resultant matrix method of two polynomials to compute a robust AGCD. The proposed work differs from other work in that it does not require any prior knowledge to recover an image from its blurred version.

Results obtained from applying linear algebra techniques to the image deblurring problem show that these techniques are more effective in removing the blurring effects from the processed images. Further research recommendations and pointers have been given in this chapter for future work. For example, considering the performance of the algebraic based BID methods in the presence of various levels of additive noise and for blurred image. The completion of these suggested areas of future work would be beneficial to gain a more accurate picture of the strengths and benefits of using algebraic methods in image deblurring methods.

Bibliography

[1] Z. Li, Z. Yang, and L. Zhi, "Blind image deconvolution via fast approximate GCD," in *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, (New York, USA), pp. 155–162, ACM, 2010.

[2] F. Li, Z. Li, D. Saunders, and J. Yu, "A theory of coprime blurred pairs," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 217–224, 2011.

[3] D. Perrone and P. Favaro, "Total variation blind deconvolution: The devil is in the details," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[4] P. Shearer, A. C. Gilbert, and A. O. Hero, "Correcting camera shake by incremental sparse approximation," in *Proceedings of IEEE International Conference on Image Processing*, pp. 572–576, 2013.

[5] S. Babacan, R. Molina, M. Do, and A. Katsaggelos, "Bayesian blind deconvolution with general sparse image priors," in *Proceedings of European Conference on Computer Vision*, vol. 7577, pp. 341–355, Springer, 2012.

[6] W. Richardson, "Bayesian-based iterative method of image restoration," *Optical Society of America*, vol. 62, no. 1, pp. 55–59, 1972.

[7] R. Lagendijk, J. Biemond, and D. Boekee, "Identification and restoration of noisy blurred images using the expectation-maximization algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 7, pp. 1180–1191, 1990.

[8] R. Gonzalez and R. Woods, *Digital Image Processing.* Upper Saddle River, New Jersey USA: Prentice-Hall, Inc., 2002.

[9] R. Lagendijk, J. Biemond, and D. Boekee, "Regularized iterative image restoration with ringing reduction," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 12, pp. 1874–1888, 1988.

[10] P. Hansen, J. Nagy, and D. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering.* Fundamentals of Algorithms, SIAM, 2006.

[11] A. Khan, *Efficient Methodologies for Single-Image Blind Deconvolution and Deblurring.* The University of Manchester, England: PhD thesis, 2014.

[12] T. Bishop, *Blind Image Deconvolution: Nonstationary Bayesian approaches to restoring blurred photos.* The University of Edinburgh: PhD thesis, 2008.

[13] H. Yin, W. Wang, and V. Rayward-Smith, *Intelligent Data Engineering and Automated Learning.* Springer-Verlag Berlin Heidelberg, 2011.

[14] J. Lee and Y. Ho, "High-quality non-blind image deconvolution," in *Proceedings of Fourth Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, pp. 282 –287, nov. 2010.

[15] S. Ramya and T. Mercy Christial, "Restoration of blurred images using blind deconvolution algorithm," in *Proceedings of International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT)*, pp. 496 –499, 2011.

[16] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *Journal of Signal Processing Magazine, IEEE*, vol. 13, no. 3, pp. 43 –64, 1996.

[17] J. Winkler, X. Lao, and M. Hasan, "The computation of a structured low rank approximation of the sylvester resultant matrix of two inexact polynomials by approximate polynomial factorisation," tech. rep., Department of Computer Science, The University of Sheffield, United Kingdom, 2000.

[18] S. Pillai and B. Liang, "Blind image deconvolution using a robust GCD approach," *IEEE Transactions on Image Processing*, vol. 8, no. 2, pp. 295–301, 1999.

[19] B. Liang and S. Pillai, "Blind image deconvolution using a robust 2-D GCD approach," in *Proceedings of IEEE International Symposium on Circuits and Systems ISCAS '97.*, vol. 2, pp. 1185–1188, 1997.

[20] E. Lopez-Rubio, R. Luque-Baena, and E. Dominguez, "Foreground detection in video sequences with probabilistic self-organizing maps," *International Journal of Neural Systems*, vol. 21, no. 3, pp. 225–246, 2011.

[21] R. Cruz-Barbosa and A. Vellido, "Semi-supervised analysis of human brain tumours from partially labelled MRS information, using manifold learning models," *International Journal of Neural Systems*, vol. 21, no. 1, pp. 17–29, 2011.

[22] D. Li and S. Simske, "Atmospheric turbulence degraded-image restoration by kurtosis minimization," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 2, pp. 244–247, 2009.

[23] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman, "Removing camera shake from a single photograph," in *Proceedings of ACM transactions on Graphics (TOG)*, vol. 25, pp. 787–794, University of Toronto, 2006.

[24] A. Agrawal, Y. Xu, and R. Raskar, "Invertible motion blur in video," in *Proceedings of ACM Transactions on Graphics (TOG)*, vol. 28, (New York, USA), p. 95, 2009.

[25] M. Ramsey and P. Christensen, "Mineral abundance determination: Quantitative deconvolution of thermal emission spectra," *Geophysical Research: Solid Earth*, vol. 103, no. B1, pp. 577–596, 1998.

[26] W. Jun and C. Danqing, "Deblurring texture extraction from digital aerial image by reforming steep edge curve," *Geo-spatial Information Science*, vol. 8, no. 1, pp. 39–44, 2005.

[27] T. Taxt and J. Strand, "Two-dimensional noise-robust blind deconvolution of ultrasound images," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 48, no. 4, pp. 861–866, 2001.

[28] S. Jefferies, K. Schulze, C. Matson, K. Stoltenberg, and E. K. Hege, "Blind deconvolution in optical diffusion tomography," *Optics Express*, vol. 10, no. 1, pp. 46–53, 2002.

[29] G. Desider, B. Anconelli, M. Bertero, P. Boccacci, and M. Carbillet, "Application of iterative blind deconvolution to the reconstruction of LBT LINC-NIRVANA images," *Astronomy and Astrophysics*, vol. 452, no. 2, pp. 727–734, 2006.

[30] J. Ligorria and C. Ammon, "Iterative deconvolution and receiver-function estimation," *Journal of Bulletin of the Seismological Society of America*, vol. 89, no. 5, pp. 1395–1400, 1999.

[31] T. Chan and C. Wong, "Total variation blind deconvolution," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 370–375, 1998.

[32] R. Lagendijk, A. Tekalp, and J. Biemond, "Maximum likelihood image and blur identification: A unifying approach," *Optical Engineering*, vol. 28, no. 5, pp. 422–435, 1990.

[33] R. Wiggins, "Minimum entropy deconvolution," *Geoexploration*, vol. 16, pp. 21 – 35, 1978.

[34] D. Kundur and D. Hatzinakos, "A novel blind deconvolution scheme for image restoration using recursive filtering," *IEEE Transactions on Signal Processing*, vol. 46, no. 2, pp. 375–390, 1998.

[35] M. Banham and A. Katsaggelos, "Digital image restoration," *IEEE on Signal Processing Magazine*, vol. 14, no. 2, pp. 24–41, 1997.

[36] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," in *Proceedings of ACM Transactions on Graphics (TOG)*, vol. 27, (New York, USA), p. 73, 2008.

[37] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *International Journal of Computer Vision*, vol. 98, no. 2, pp. 168–186, 2012.

[38] A. Danelakis, M. Mitrouli, and D. Triantafyllou, “Blind image deconvolution using a banded matrix method,” *Numerical Algorithms*, pp. 1–30, 2012.

[39] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising with block-matching and 3D filtering,” in *Proceedings of International Society for Optics and Photonics on Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, vol. 6064, p. 606414, 2006.

[40] A. Foi, V. Katkovnik, and K. Egiazarian, “Pointwise shape-adaptive DCT for high-quality deblocking of compressed color images,” in *Proceedings of 14th European on Signal Processing Conference*, pp. 1–5, 2006.

[41] G. Ayers and J. Dainty, “Iterative blind deconvolution method and its applications,” *Optics letters*, vol. 13, no. 7, pp. 547–549, 1988.

[42] J. Nagy, K. Palmer, and L. Perrone, “Iterative methods for image deblurring: A matlab object-oriented approach,” *Numerical Algorithms*, vol. 36, no. 1, pp. 73–93, 2004.

[43] A. Veen and A. Paulraj, “An analytical constant modulus algorithm,” *IEEE Transactions on Signal Processing*, vol. 44, no. 5, pp. 1136–1155, 1996.

[44] L. He, A. Marquina, and S. Osher, “Blind deconvolution using TV regularization and bregman iteration,” *International Journal of Imaging Systems and Technology*, vol. 15, no. 1, pp. 74–83, 2005.

[45] R. Kennedy and P. Samarasinghe, “Efficient blind separable kernel deconvolution for image deblurring,” in *Proceedings of 2nd International Conference on Signal Processing and Communication Systems*, pp. 1–7, 2008.

[46] J. Shi and S. Reichenbach, “Image interpolation by two-dimensional parametric cubic convolution,” *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1857–1870, 2006.

[47] C. Lee and W. Hwang, “Mixture of Gaussian blur kernel representation for blind image restoration,” *IEEE Transactions on Computational Imaging*, vol. PP, no. 99, pp. 1–1, 2017.

[48] C. Lee and W. Hwang, “Sparse representation of a blur kernel for out-of-focus blind image restoration,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp. 2698–2702, 2016.

[49] S. Reichenbach and F. Geng, “Two-dimensional cubic convolution,” *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 857–865, 2003.

[50] L. Fang, H. Liu, F. Wu, X. Sun, and H. Li, “Separable kernel for image deblurring,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2885–2892, 2014.

[51] N. Alkhaldi, *Blind Image Deconvolution Using The Sylvester Matrix*. The University of Sheffield, United Kingdom: PhD thesis, 2014.

[52] G. Ayers and J. Dainty, “Iterative blind deconvolution method and its applications,” *Optics letters*, vol. 13, no. 7, pp. 547–549, 1988.

[53] B. Hunt, “The application of constrained least squares estimation to image restoration by digital computer,” *IEEE Transactions on Computers*, vol. C-22, no. 9, pp. 805–812, 1973.

[54] R. Lagendijk, J. Biemond, and D. Boekee, “Identification and restoration of noisy blurred images using the expectation-maximization algorithm,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 7, pp. 1180–1191, 1990.

[55] T. Chan and C.-K. Wong, “Total variation blind deconvolution,” *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 370–375, 1998.

[56] A. Levin, Y. Weiss, F. Durand, and W. Freeman, “Understanding and evaluating blind deconvolution algorithms,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1964–1971, 2009.

[57] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," in *Proceedings of ACM transactions on Graphics (TOG)*, vol. 27, (New York, USA), p. 73, 2008.

[58] S. Cho and S. Lee, "Fast motion deblurring," in *Proceedings of ACM Transactions on Graphics (TOG)*, vol. 28, p. 145, 2009.

[59] A. Levin, Y. Weiss, F. Durand, and W. Freeman, "Efficient marginal likelihood optimization in blind deconvolution," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2657–2664, 2011.

[60] L. Xu, S. Zheng, and J. Jia, "Unnatural L0 sparse representation for natural image deblurring," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1107–1114, 2013.

[61] X. Zhou, J. Mateos, F. Zhou, R. Molina, and A. Katsaggelos, "Variational dirichlet blur kernel estimation," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5127–5139, 2015.

[62] A. Hyvärinen, "Independent component analysis: recent advances," *Mathematical, physical, and engineering sciences*, 2013.

[63] A. Hyvärinen and E. Oja, "Independent Component Analysis: Algorithms and Applications," *Journal of Neural Netw.*, vol. 13, pp. 411–430, 2000.

[64] P. Comon, "Independent Component Analysis, a New Concept," *Special issue on High-Order Statistics*, vol. 36, no. 3, pp. 287–314, 1994.

[65] P. Comon, "Independent Component Analysis, Higher Order Statistics," *the International Signal Processing Workshop*, pp. 29–38, 1991.

[66] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, vol. 46. A Wiley Interscience Publication John Wiley and Sons, 2004.

[67] H. Yin and I. Hussain, "Independent component analysis and nongaussianity for blind image deconvolution and deblurring," *Integrated Computer-Aided Engineering*, vol. 15, p. 219–2 219, 2008.

[68] S. Umeyama, "Blind deconvolution of images using Gabor filters and independent component analysis," in *Proceedings of Symp. Independent Component Analysis and Blind Signal Separation*, (Nara, Japan), pp. 319–324, ICA, 2003.

[69] S. Umeyama, "Blind deconvolution of blurred images by use of ICA," *Journal of Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 84, no. 12, pp. 1–9, 2001.

[70] M. Welling, "Robust higher order statistics.," in *Proceedings of AISTATS*, no. 3, p. 7, 2005.

[71] L. Albera, A. Ferreol, P. Chevalier, and P. Comon, "ICAR: a tool for blind source separation using fourth-order statistics only," *IEEE Transactions on Signal Processing*, vol. 53, pp. 3633 – 3643, 2005.

[72] A. Hyvärinen, P. O. Hoyer, and E. Oja, "Sparse code shrinkage: Denoising by nonlinear maximum likelihood estimation," in *Proceedings of Advances in Neural Information Processing Systems*, vol. 11(7):1739–1768, pp. 473–479, 1999.

[73] Q. Zhang, H. Yin, and N. M. Allinson, "A simplified ICA based denoising method," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 5, pp. 479–482, 2000.

[74] A. Bell and T. Sejnowski, "The "independent components" of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327 – 3338, 1997.

[75] A. Khan and H. Yin, "Spectral non-gaussianity for blind image deblurring," in *Proceedings of International Conference on Intelligent Data Engineering and Automated Learning*, pp. 144–151, Springer, 2011.

[76] A. Khan and H. Yin, "Quality measures for blind image deblurring," in *Proceedings of IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 456–459, 2012.

[77] A. Heindl, *Fourier transform, polynomial GCD, and image restoration.* Master Thesis, 2005.

[78] S. Barnett, *Polynomials and linear control systems (Pure and applied mathematics).* New York: Marcel Dekker, 1983.

[79] G. Golub and C. V. Loan, *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition).* The Johns Hopkins University Press, 3rd ed., 1996.

[80] M. Hasan, *The computation of multiple roots of a polynomial using structure preserving matrix methods.* The University of Sheffield, Sheffield, England: PhD thesis, 2011.

[81] X. Lao, *Structured matrix methods for a polynomial root solver using approximate greatest common divisor computations and approximate polynomial factorisations.* The University of Sheffield, Sheffield, England: PhD thesis, 2011.

[82] J. Winkler and X. Lao, "The calculation of the degree of an approximate greatest common divisor of two polynomials," *Computational and Applied Mathematics*, vol. 235, no. 6, pp. 1587–1603, 2011.

[83] R. Corless, P. Gianni, B. Trager, and S. Watt, "The singular value decomposition for polynomial systems," in *Proceedings of the international symposium on Symbolic and algebraic computation*, ISSAC '95, (New York, USA), pp. 195–207, ACM, 1995.

[84] J. Winkler and M. Hasan, "A non-linear structure preserving matrix method for the low rank approximation of the sylvester resultant matrix," *Computational and Applied Mathematics*, vol. 234, no. 12, pp. 3226 – 3242, 2010.

[85] J. Winkler and J. Allan, "Structured total least norm and approximate GCDs of inexact polynomials," *Computational and Applied Mathematics*, vol. 215, no. 1, pp. 1 – 13, 2008.

[86] D. Dunaway, "Calculation of zeros of a real polynomial through factorization using euclid's algorithm," *SIAM Journal on Numerical Analysis*, vol. 11, no. 6, pp. 1087–1104, 1974.

[87] S. Ghaderpanah and S. Klasa, "Polynomial scaling," *SIAM Journal on Numerical Analysis*, vol. 27, no. 1, pp. 117–135, 1990.

[88] J. R. Winkler, M. Hasan, and X. Lao, "Two methods for the calculation of the degree of an approximate greatest common divisor of two inexact polynomials," *Calcolo*, vol. 49, no. 4, pp. 241–267, 2012.

[89] R. Goldman, T. Sederberg, and D. Anderson, "Vector elimination: A technique for the implicitization, inversion, and intersection of planar parametric rational polynomial curves," *Computer Aided Geometric Design*, vol. 1, no. 4, pp. 327 – 356, 1984.

[90] T. Sederberg and G. Chang, "Best linear common divisors for approximate degree reduction," *Computer-Aided Design*, vol. 25, no. 3, pp. 163 – 168, 1993.

[91] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint.* Cambridge, USA: MIT Press, 1993.

[92] S. Petitjean, "Algebraic geometry and computer vision: Polynomial systems, real and complex roots," *Mathematical Imaging and Vision*, vol. 10, no. 3, pp. 191–220, 1999.

[93] J. Kajiya, "Ray tracing parametric patches," *SIGGRAPH Comput. Graph.*, vol. 16, no. 3, pp. 245–254, 1982.

[94] V. Hribernig and H. Stetter, "Detection and validation of clusters of polynomial zeros," *Symbolic Computation*, vol. 24, no. 6, pp. 667 – 681, 1997.

[95] R. Corless, P. Gianni, B. Trager, and S. Watt, "The singular value decomposition for polynomial systems," in *Proceedings of the 1995 international symposium on Symbolic and algebraic computation*, pp. 195–207, ACM, 1995.

[96] R. Corless, S. Watt, and L. Zhi, "QR factoring to compute the GCD of univariate approximate polynomials," *IEEE Transactions on Signal Processing*, vol. 52, no. 12, pp. 3394–3402, 2004.

[97] J. Winkler, X. Lao, and M. Hasan, "The computation of multiple roots of a polynomial," *Computational and Applied Mathematics*, vol. 236, no. 14, pp. 3478–3497, 2012.

[98] M. Bourne, J. Winkler, and S. Yi, “The computation of the degree of an approximate greatest common divisor of two bernstein polynomials,” *Applied Numerical Mathematics*, vol. 111, pp. 17 – 35, 2017.